NASA CR-112297

AN ANALYTICAL STUDY FOR THE DESIGN OF ADVANCED ROTOR AIRFOILS

BY Larry D. Kemp

BELL HELICOPTER COMPANY REPORT NO. 299-099-635

March 29, 1973

Prepared Under Contract No. NASW-2334 by
Bell Helicopter Company, a Textron Company
Fort Worth, Texas

for

NATIONAL AERONAUTICS AND SPACE ADMINISTRATION
LANGLEY RESEARCH CENTER

ABSTRACT

A theoretical study has been conducted to design and evaluate two airfoils for helicopter rotors. The best basic shape, designed with a transonic hodograph design method, was modified to meet subsonic criteria. One airfoil had an additional constraint for low pitching-moment at the transonic design point. Airfoil characteristics were predicted. Results of a comparative analysis of helicopter performance indicate that the new airfoils will produce reduced rotor power requirements compared to the NACA 0012.

The hodograph design method, written in CDC Algol, is listed and described.

ACKNOWLEDGEMENT

The author wishes to express his appreciation to the NLR personnel for their fine work associated with the development of the two airfoil sections and to Messrs. Jan M. Drees and John M. Duhon for their technical guidance and comments.

SUMMARY

The work presented in this report was done under NASA Contract NASW-2334. This contract was awarded to Bell Helicopter Company (BHC) for an analytical study with the objective to define two specific airfoil shapes for helicopter rotors. These shapes were to be designed to satisfy two transonic design requirements established by the high speed forward and maneuvering flight conditions without compromising the subsonic requirements dictated by the hovering condition.

BHC subcontracted the design portion of the contract to the National Aerospace Laboratories (NLR) in the Netherlands. NLR combined their transonic hodograph design method with the multiple requirement design approach developed by Dr. F. X. Wortmann for BHC. Two airfoil sections were developed and their corresponding subcritical, aerodynamic properties were predicted. One airfoil had a maximum allowable pitching moment coefficient, while the other had no such restriction. BHC then added to the aerodynamic data supplied by NLR by estimating comparable data for the supercritical flight conditions. Using the combined data, performance calculations were made for hover, high speed forward flight, and maneuvering flight. All performance calculations exhibited an improvement over a conventional, contemporary rotor blade section for all three flight regimes.

The report contains recommendations for further airfoil optimization and future experiments to verify the analytical predictions.

TABLE OF CONTENTS

TABLE OF CONTENTS (Cont'd)

LIST OF SYMBOLS

c	Chord - meter
c_ℓ	Lift coefficient
c_d	Drag coefficient
c_m	Moment coefficient
M_a	Mach number
M_{TIP}	Tip Mach number
C_T	Thrust coefficient $(T/(\rho\pi R^2 (\Omega R)^2))$
C_P	Power coefficient $(P/(\rho\pi R^2 (\Omega R)^3))$
g	Gravitational loading
P	Rotor power
P_E	Equivalent power
R	Rotor radius - meter
T	Rotor thrust - newtons
n	Number of blades
α	Angle of attack - degrees
ϵ	Thickness control parameter
λ_1	Leading edge bluntness control parameter
λ_2	Leading edge droop control parameter
ρ	Sea level standard atmospheric density (.05979 Kg/m^3)
σ	Rotor solidity $(nc/\pi R)$
Ω	Rotation speed - rad/sec

LIST OF TABLES

LIST OF ILLUSTRATIONS

LIST OF ILLUSTRATIONS (Cont'd)

Page Intentionally Left Blank

I. INTRODUCTION

In recent years it has become evident that the design requirements for helicopter rotor blade airfoil sections differ enough from those of fixed-wing aircraft to justify an independent development program. As a result, analytically designed airfoils tailored to optimize hover, maneuver, and high speed performance simultaneously are now in use and have been tested on full-scale rotors (BHC Airfoil Section FX69-H-098).

Contractor efforts in this field were greatly accelerated as a result of the contributions by Dr. F. X. Wortmann (Professor at Stuttgart University), consultant to Bell Helicopter Company (BHC). Dr. Wortmann's design approach requires specification of key operating conditions for which optimum performance is sought. Those operating conditions are then expressed as specific design points (see Table I). The computerized airfoil design method is then used to determine the incompressible, viscous, velocity distributions for the given set of design conditions. By determining the elements of the airfoil contour principally involved in attaining the design inputs, an airfoil can usually be found that will excel at each of the desired design conditions. Figure 1 shows a typical example of an airfoil designed in this manner.

Incompressible flow fields were used to achieve the above results. However, the high g maneuver (high c_ℓ, moderate M_a) and the high-speed forward flight (low c_ℓ, high M_a) requirements involved analysis in the transonic speed range. Dr. Wortmann supplemented his above analysis with the "peaky" approach developed by Pearcey (Ref. 1). With this method, he used the principle of the "peaky" pressure distribution for both of the transonic design conditions to reduce the strength of the shockwaves.

This study was conducted as a continuation to the preceding study. One of the main objectives of this program was to try to further reduce or even eliminate the shockwaves at the two transonic design points by using advanced computational methods for transonic flow. In order to do this under NASA contract, BHC subcontracted the National Aerospace Laboratories (NLR) to develop two airfoil sections using their hodograph technique for quasi-elliptical airfoils. BHC assisted NLR by supplying detailed information concerning the method used by Dr. Wortmann and the airfoil he developed (FX69-H-098), as well as the design requirements for the two new sections (see Table II).

The specific objectives of this study are fourfold:

(1) Determine if it is possible to design, analytically, an airfoil that fulfills the two transonic design requirements (maneuver and high speed flight) of Table II while simultaneously satisfying stringent subsonic requirements (hover). And, if this is possible, develop two sections that both satisfy the same transonic requirements, but with one having a restricted pitching moment.

(2) Determine if the transonic hodograph theory for lifting quasi-elliptical airfoils is a convenient tool for achieving objective (1).

(3) Determine the aerodynamic performance penalties incurred by the requirement for small pitching moment coefficients usually imposed for rotary wing airfoils.

(4) Predict the rotor performance benefits that may be expected from the improved airfoil sections in hover, high-speed flight, and maneuvering flight when compared to the performance produced by a rotor having an NACA 0012 or an FX69-H-098 section.

It is felt that some additional comments are needed concerning objectives (3) and (4). For objective (3), it is stressed that the penalties incurred by the pitching moment requirements are not to be analyzed from a control load standpoint, but rather from a performance standpoint. For objective (4), it is noted that all performance predictions are for the same mathematical rotor model with changes occurring only in the airfoil data. The mathematical model used a rigid blade, teetering type rotor system. All performance should be considered from a "relative" rather than an "absolute" point of view.

II. SUMMARY OF NLR WORK

A. Introduction

The following is a brief discussion of the work performed by NLR. The complete discussion and analysis are found in Appendix A, "An Aerodynamic Design Study for Rotor Airfoils", and Appendix B, "ALGOL Programs for the Computation of Quasi-Elliptical Shock-Free Transonic Aerofoils". In summary, the technique used by NLR consisted of first using the hodograph method to obtain a series of shock-free shapes, select from this series the shape that appears most promising with respect to the other requirements, and then modifying this shape by means of a trial-and-error method to further optimize for other requirements.

B. Airfoil Development

Following the receipt of the design requirements (see Table II), as well as the data and information concerning the methods used by Dr. F. X. Wortmann in developing the FX69-H-098 section, NLR began design of the required sections. Initially, two approaches were considered. The first consisted of developing a basic section that satisfied the high Mach number - low c_ℓ requirement and then modifying this section to improve the high c_ℓ- moderate Mach number requirements. The second approach was to develop the high c_ℓ - moderate Mach number section and modify this section to improve its high Mach number - low c_ℓ characteristics.

Work was performed using both methods; however, it was soon concluded that the first approach was the most desirable. By selecting this method, the hodograph program was given the responsibility of developing the larger portions of the sections. This left the smaller segment of the section to be modified by hand-fitting techniques.

All of the above work was done using the hodograph program with the four basic input parameters: (1) Mach number control (M_a), (2) Circulation control (Γ), (3) Angle-of-attack control (α), and (4) Thickness control (ε). A basic section, designated the NLR 7216 section, was developed using the program and these control parameters. It was soon realized from analyzing the pressure field and surface curvature distribution, however, that more leading-edge droop would be needed to satisfy the maximum c_ℓ requirement. In order to achieve this, two additional control parameters, λ_1 and λ_2, the values of which were so far set equal to zero, were made operational. The first parameter controls the leading edge bluntness and the second controls the leading-edge droop. With these modifications, a satisfactory basic section (NLR 7223) was developed.

The final two sections (NLR 7223-62 and NLR 7223-43) shown in Figure 2 evolved after considerable modifications were attempted using trial-and-error techniques. Each attempt was checked by both subsonic potential and viscous flow calculations in addition to other empirical methods. These hand-fitting techniques yielded sections that were modified in the following approximate areas:

Upper Surface:	0 to 0.02 x/c
	0.70 to 1.00 x/c
Lower Surface:	0.60 to 1.00 x/c

Both airfoil sections were obtained as described above. Different contours and velocity fields resulted, however, for the two sections due to the different pitching moment requirements. In designing both sections, care had to be exercised that the resulting velocity field would not endanger the boundary layer development on both the upper and lower surfaces and possibly cause premature separation.

C. Aerodynamic Data Calculations

Once the development of the two sections was complete, aerodynamic data were calculated and estimated for the off-design conditions. For these conditions, where attached flow was believed to exist, NLR used several methods combined in a single computer program (Reference 23*). This program calculates a potential flow field using the method of Reference 18*. The flow field is then combined with the boundary layer calculation methods of references 20*, 21*, and 22*. The drag values are then calculated using the method described in Reference 24*. All the above calculations are limited to subcritical, fully-attached flows.

NLR used two procedures combined with the FX69-H-098 experimental data for estimating maximum lift coefficients. Both methods provided estimated, incremental maximum lift coefficient values that were used with the experimental data.

Both methods were based on and limited by the assumption that the stall mechanism for the new sections was similar to that of the FX69-H-098 airfoil. For the first method incremental coefficient values were estimated by noting the relation of minimum pressure as a function of lift coefficient at the critical pressure value (see Figure 15*). For the second method incremental values were determined by utilizing Sinnott's cirterium (Reference 25*) in relation to the crest pressure expressed as a function of lift coefficient (see Figure 16*).

Table III shows a summary of both the design objectives and the values claculated, or estimated, by NLR.

* Appendix A reference number

III. AERODYNAMIC SECTION DATA

A. Presentation of Data

c_ℓ-α, c_d-α, and c_ℓ-c_d data are shown in Figures 3 through 14 for the NACA 0012, FX69-H-098, NLR 7223-62, and NLR 7223-43 airfoil sections. For convenience, these sections will be referred to as the 0012, 098, Airfoil 1, and Airfoil 2 sections, respectively.

Three types of data are shown in these figures: experimental data (0012 and 098 sections), NLR data (Airfoil 1 and Airfoil 2 sections), and BHC data (all sections). Where discrepancies exist between NLR and BHC data, NLR data was used in the performance calculations. Likewise, the test data always took precedence over any calculated data.

1. Experimental Data. The experimental data for the 0012 section were obtained from Reference 2, and that for the 098 section from tests conducted by BHC at the United Aircraft Research Laboratories (Reference 3). These data were used in evaluating the calculation methods and in performance predictions.

2. NLR Data. The NLR data shown in Figures 9 through 14 were derived from both calculations and estimations with no distinction being made as to which was used. Figures 20 through 23 and 39 through 42 of Appendix A show this data in more detail. For a complete discussion on the methods and techniques used for determining these data see Section II of this report and Section 6 of Appendix A.

3. BHC Data. Aerodynamic data were calculated and estimated by BHC using several different methods. These data were produced to supplement the experimental data as well as the NLR calculated data beyond Mach number and angle-of-attack values that were available to BHC from test or NLR predictions. Consequently, all BHC data were made to "fair-in" to the NLR or test data.

B. Calculations

1. Basic Method

The BHC basic method adds the c_d-c_ℓ results calculated from two computer programs. One program was developed for BHC by Dr. F. X. Wortmann and the other program was developed by Bauer, Garabedian, and Korn (Reference 4). Dr. Wortmann's program utilizes an incompressible, two-dimensional flow for calculating flow fields around a given section. Boundary layer calculations are then made with transition occurring at the position where laminar separation would normally result for a Reynolds number of 5×10^6.

Only the "off-design" or analysis portion of the Bauer, Garabedian, and Korn transonic flow program was used.* This program analyzes a given section in a two-dimensional, compressible, potential flow field. The resulting wave drag as a function of c_ℓ was added to the incompressible viscous drag obtained from the Wortmann program to obtain the total drag coefficient. The accuracy of this method is considered to be very good as shown in Figures 8, 11, and 14. The data calculated using this method agree very well with both the experimental data and the NLR calculated data.

2. Alternate Methods

The BHC basic method as described above was used as long as the transonic flow program could obtain convergence. Once the program failed to converge for a given Mach number/angle-of-attack combination, then empirical methods had to be used. These methods are discussed below relative to the airfoils for which they were used.

(a) 0012 Section. c_d - α data were obtained for M_a equal to 1.0 by applying an incremental value to the data of Reference 2. These incremental data were obtained from unpublished 0012 test data at Mach numbers of 0.9 and 1.0.

It was assumed that the c_ℓ - α relation remained unchanged for M_a values between 0.9 and 1.0.

(b) 098 Section. c_d - c_ℓ data for M_a values between 0.82 and 0.89 were calculated by the BHC basic method. These data were extrapolated to M_a equal to 0.9. A similar extrapolation for minimum c_d is shown in Figure 15. The c_d - c_ℓ data for M_a equal to 1.0 were estimated by assuming the same incremental values as were used for the 0012 section.

c_ℓ - α data for M_a equal to 0.9 were obtained by extrapolating the test data as shown in Figure 16. Again it was assumed that the c_ℓ - α relation remained unchanged for M_a between 0.9 and 1.0.

Estimates for maximum c_ℓ for $M > 0.7$ are shown in Figure 17. It is believed that this extrapolation yielded conservative results.

(c) NLR Sections. The same procedure was used for estimating the c_d - c_ℓ relation for these sections as was used for the 098 section for M_a values through 0.9 (typical data are shown in Figure 15). For M_a equal to 1.0 however, the c_d - c_ℓ values for the 098 section were used rather than adding the 0012 incremental values to the NLR data at M_a equal to 0.9. It was believed that this incremental method would have yielded too low of values for c_d when applied to the NLR sections.

*All calculations using this program were made with a crude grid mesh size and with the artificial viscosity parameter equal to zero.

The c_ℓ-α data for M_a values between 0.7 and 0.9 were estimated by extrapolating the data as shown in Figure 16, assuming that similar trends existed between the 098 and NLR sections. The same c_ℓ-α was assumed for both M_a equal to 0.9 and 1.0.

Similar assumptions were made for the extrapolation of the maximum c_ℓ data shown in Figure 17.

IV. PERFORMANCE ANALYSIS

A. Introduction and Assumptions

Performance calculations were made for three flight modes: (1) hover, (2) high-speed forward flight, and (3) a steady-state pull-up maneuver. All performance calculations were made using two BHC computer programs, ARSF03* and AGAJ68**. The hover and level flight performance was calculated with the first program and the maneuvering performance with the second.

ARSF03 employs blade-element-momentum theory with non-uniform inflow for hover and axial flight, and uniform inflow for the other flight conditions. The effects of stall, compressibility, and reverse flow are determined by utilizing two-dimensional airfoil data which specify aerodynamic characteristics throughout the angle-of-attack and Mach number range. Geometric characteristics are also specified at a given number of radial blade stations. Reference 5 provides further discussion of the theory.

AGAJ68 is a BHC rotorcraft flight simulation analysis program and was used to simulate maneuvering flight conditions. Essentially, the program consists of a rotor aerodynamic and dynamic analysis coupled with a fuselage analysis which includes all six rigid-body degrees of freedom. Detailed descriptions of this program can be found in Reference 6.

All calculations were made for sea level, standard day conditions along with the following rotor parameters:

1. Radius = 7.62 meters (25 feet)
2. Solidity = 0.07
3. Twist = -8.0 degrees
4. Lock No. = 7.0

The NLR and BHC calculated aerodynamic section data, as shown in Figures 3 through 14, were used for the performance analysis. For angles of attack greater than 16 degrees and less than -4 degrees (i.e., the reverse flow region) 0012 section data (Reference 7) were used and are shown in Figure 18. No compressibility effects were applied to these section data.

A performance summary is found in Table IV for a typical 62 275 newtons (14 000-pound) class vehicle with 1.39 square meters (15 square feet) flat plate drag area and a rotor tip speed of 234.7 meters per second (770 feet per second).

B. Hover Performance

Figures 19 through 21 show hover performance for all four airfoil sections. These data were calculated for a thrust range of 35 586 to 80 068 newtons (8000 to 18 000 pounds) and hovering tip speeds of 226, 235, and 244 meters per second (740, 770, and 800 feet per second).

* Formally known as the BHC F35 computer program
** Formally known as the BHC C81 computer program

Assuming that a typical design hovering C_T/σ is approximately 0.065, it may be concluded that the 098 and NLR sections would yield approximately the same hovering performance. A power savings of approximately eleven percent would be realized over a rotor using the 0012 section. In dimensional form, this would be a savings of about 105 kw (140 hp), or a 6094 nt (1370 lb) increase in thrust. It should be noted for the hovering performance, and all other quoted performance, that the quoted values are a function of rotor tip Mach number. Care should be taken in applying these values to other design conditions.

C. High-Speed Performance

Figures 22 through 24 show predicted high-speed forward flight performance. For these calculations, an airframe flat plate drag area of 1.39-square meters (15-square feet) and a gross weight of 62 275 newtons (14 000 pounds) was assumed. Again, rotor tip speeds of 226, 235, and 244 meters per second (740, 770, and 800 feet per second) were used, and data were calculated from 80 to 180 knots. As noted for a rotor tip speed of 235 meters per second (770 feet per second) and 1268 kilowatts power (1700 hp), an increase of 11, 20, and 23 knots can be realized over the 0012 section for the 098, Airfoil 2, and Airfoil 1 sections, respectively. Or, for a cruise speed of 150 knots, a fuel savings of 17, 28, and 33 percent would result.

D. Maneuvering Performance

Maneuver performance is shown in Figure 25 for a steady state pull-out type maneuver at 150 knots. The data are shown as normalized equivalent horsepower* (Reference 8), versus the load factor. The horsepower data has been normalized to the 0012 data in the cruise condition. As shown, both the 098 and NLR sections show an improvement over the 0012 with an increase of eight to eleven percent. This improvement seems to be nearly independent of load factor once the retreating blade begins to enter deep stall. As an example of retreating blade stall characteristics, it is noted that for a load factor of 1.8 that the entire retreating blade is beyond 20 degrees angle of attack for all four blade sections analyzed.

*Equivalent horsepower is the total horsepower supplied to the rotor whether by engine or the conversion of kinetic or potential energy to rotor power.

V. CONCLUSIONS

Two new helicopter rotor airfoil sections have been developed through a joint effort between NASA, NLR, and BHC. Concerning the four objectives of this study, the following conclusions have been made.

Objective 1:

It is possible to design an airfoil analytically that fulfills the requirements for two transonic design points while satisfying stringent subsonic requirements. The two sections produced by NLR satisfy all the requirements except maximum c_ℓ as shown in Table III.

Objective 2:

The transonic hodograph theory for lifting quasi-elliptical airfoils is a convenient tool for the design of such sections as described above. It was determined by NLR, however, that some "hand-fitting" of the basic shapes produced by the hodograph theory had to be done. This is explained completely in Appendix A. Basically, a leading edge modification was applied with the objectives of increasing maximum c_ℓ, and a trailing edge modification for maintaining laminar flow and/or attached flow depending upon which surface was being modified. The trailing edge modification was also used for controlling the pitching moment.

Objective 3:

It was found that the only apparent penalty incurred by an airfoil section requiring a low pitching moment coefficient (Airfoil 1) is a slight reduction in maximum lift coefficient (1.30 to 1.25). It was also noted that a potential gain may even exist for the low moment type section. This conclusion is based upon both NLR and BHC calculated c_d data at $c_\ell < 0.2$. Both sets of data show the low moment section to have a higher drag divergence Mach number. This fact may be clearly seen when the data in Figures 11 and 14 are compared for $M_a = 0.9$. Also, the low moment section is showing a slightly lower value of c_d in the range of $0.3 < c_\ell < 0.6$ for $M_a < 0.6$.

From a performance standpoint, the two sections yielded nearly the same results with the high moment section (Airfoil 2) yielding slightly better results in the maneuvering flight mode. In the high speed flight mode a sufficient amount of the rotor had been subjected to the higher Mach number range (>0.9) in order for the improved Mach number characteristics of the low moment section to be felt (see Figures 23 and 24).

It was also noted that the low moment section yielded slightly better hovering performance as seen in Table IV. This is due to the lower values of c_d in the range of $0.3 < c_\ell < 0.6$ for $M_a < 0.6$.

It is pointed out that care should be taken when comparing these or any other sections. A slight change in rotor tip Mach number, design lift point, or any number of other design requirements may cause a reversal of these conclusions.

Objective 4:

The differences in performance between the two NLR sections are described above. Both of these sections show considerable improvement over the 0012 section. When compared to the 098 section the NLR sections show comparable or better performance in all three flight regimes.

An additional fact has been discovered as a result of this study. It was noted by NLR that a characteristic "bump" resulted on the upper surface of the high speed sections. "It is believed that the presence of such a curvature peak is an essential feature of airfoils that must combine high speed and high maximum c_ℓ performance in the way required for application in a helicopter rotor." For a more complete discussion, see Appendix A.

VI. RECOMMENDATIONS

The following recommendations are made as a result of this study.

- Conduct additional analytical studies to determine if further optimization of the NLR sections is possible with the two new control parameters (λ_1 and λ_2) and to explore in detail the influence of the upper surface curvature peak height and location (see Appendix A).

- Conduct steady state two dimensional transonic-tunnel tests to determine the accuracy of the prediction methods used in this study. These tests could include measurements on a model in a yawed condition to determine the effect or sensitivity of these high speed sections to asymmetric flow conditions.

- Conduct two-dimensional oscillating transonic tests to determine the sensitivity of this type of high speed sections to unsteady flow conditions.

- Conduct rotational tests on a tail rotor size model to determine the section properties of the new airfoils under rotating conditions.

REFERENCES

1. Pearcey, H. H., The Aerodynamic Design of Section Shapes for Swept Wings, Advances in Aeronautical Sciences, (Vols. 3-4), Pergamon, London, 1962.

2. Tanner, Watson H., Charts for Estimating Rotary Wing Performance in Hover and at High Forward Speed, NASA CR-114, 1964.

3. Wind Tunnel Tests of Bell Two-Dimensional Airfoil Models, United Aircraft Research Laboratories Report J930980-1, United Aircraft Corporation, East Hartford, Connecticut, 1970.

4. Bauer, F., Garabedian, P., and Korn, D., Lecture Notes in Economics and Mathematical Systems - 66 Supercritical Wing Sections, Springer-Verlag, Berlin . Heidelberg . New York 1972.

5. Livingston, C. L., Rotor Aerodynamic Characteristics Program F35(J), Bell Helicopter Company Report 599-004-900, Bell Helicopter Company, Fort Worth, Texas, June, 1967.

6. Bird, B. J., McLarty, T. T., Livingston, C. L., A Stability and Control Prediction Method for Helicopters and Stoppable Rotor Aircraft, Vol. I, II, III, & IV, Technical Report AFFDL-TR-69-123, Air Force Flight Dynamics Laboratory, Wright-Patterson Air Force Base, Ohio, 1970.

7. Cretzos, C. C., Heyson, H. H., and Boswinkle, R. W., Jr., Aerodynamic Characteristics of NACA 0012 Airfoil Section at Angles of Attack From 0° to 180°, NACA TN3361, January 1955.

8. Wells, C. D., Wood, T. L., Maneuverability-Theory and Application, presented at the 28th National Forum of the American Helicopter Society, Washington, D. C., May, 1972.

TABLE I.

TYPICAL CONDITIONS FOR ROTOR AIRFOIL DESIGNS

	OPERATING CONDITION	AIRFOIL DESIGN POINT	SPECIFIED GOAL *
Hover		$M = .6$ $c_\ell = .65$	$c_\ell / c_d = 100$
(Max Gross, Critical alt & temp)		$M = .6$ $c_\ell = .65$ (Conditions at 3/4R)	$c_{m_{1/4}} < \lvert .02 \rvert$
Transonic	High g - Maneuver	$M \approx .5$ (Retreating blade stall)	$c_{\ell_{max}} > 1.25$
Transonic	High speed cruise	$c_\ell \approx 0$ (advancing blade tip)	$M_{drag\ rise} > .80$

* Example, pertaining to the conditions originally used to design the FX 69-H-098 airfoil

TABLE II.

DESIGN REQUIREMENTS FOR NEW AIRFOILS

FLIGHT CONDITION	SPECIFIC	QUANTITY	AIRFOIL 1 (low c_m)	AIRFOIL 2 (no c_m req)
Hover	$M = .6$ and $c_\ell = .65$	c_ℓ / c_d	100	100
	$M = .6$ and $c_\ell = .65$	$c_{m_{1/4c}}$	$< \lvert 0.02 \rvert$	no requirement
Maneuver	$M \simeq .5$	$c_{\ell_{max}}$	> 1.35 Shock free*	> 1.35 Shock free*
High Speed	$c_\ell \simeq 0$	M	> .85 Shock free*	> .85 Shock free*
		c_d	< .013	< .013
General	2-D Test Condition	Re	5×10^6	5×10^6
	Thickness Ratio	% Chord	> 4 <15	> 4 <15

* This indicates the method used to obtain the listed design objective

TABLE III.

AERODYNAMIC DATA SUMMARY

FLIGHT CONDITION	SPECIFIC	QUANTITY	DESIGN OBJECTIVES (AIRFOIL 1)	MAXIMUM OBTAINED (AIRFOIL 1)	DESIGN OBJECTIVES (AIRFOIL 2)	MAXIMUM OBTAINED (AIRFOIL 2)
Hover	$M=.6,\ c_\ell=.65$	c_ℓ/c_d	100	95	100	100
	$M=.6,\ c_\ell=.65$	$(c_M)_{C/4}$	$<\lvert .02\rvert$	-.015	no requirements	-.046
Maneuver	$M\approx.5$	$c_{\ell_{max}}$	> 1.35	1.25	> 1.35	1.30
High Speed	$c_\ell\approx 0$	M	> .85	.85	> .85	.85
		c_d	< .013	.013	< .013	.013
General	2-D Test Condition	Re	5×10^6	5×10^6	5×10^6	5×10^6
	Thickness Ratio	% Chord	$4 < t/c < 15$	8.6	$4 < t/c < 15$	8.6

TABLE IV.

PERFORMANCE SUMMARY*

PERFORMANCE PARAMETER	COMPARISONS RELATIVE TO 0012 SECTION: 098	AIRFOIL 1	AIRFOIL 2
Hover			
Hover Power Savings (%) Conditions: $C_{T/\sigma} = .065$	11.1	11.4	11.1
High Speed			
(1) Increase in speed (knots) Conditions: 1268 KW (1700 hp)	11	23	20
(2) Fuel savings (%) Conditions: 150 knots	17.4	33.2	27.8
Maneuver			
Increase load factor (%) Conditions: (1) Power for a load factor of 1.60 with 0012 section (2) 150 knots	9.5	8.5	11.2

*Basic Flight Conditions: Gross Weight = 62,272 nt (14000 lb)
Flat Plate Drag Area = 1.39 m² (15 ft²)
ΩR = 235 m/sec (770 fps)

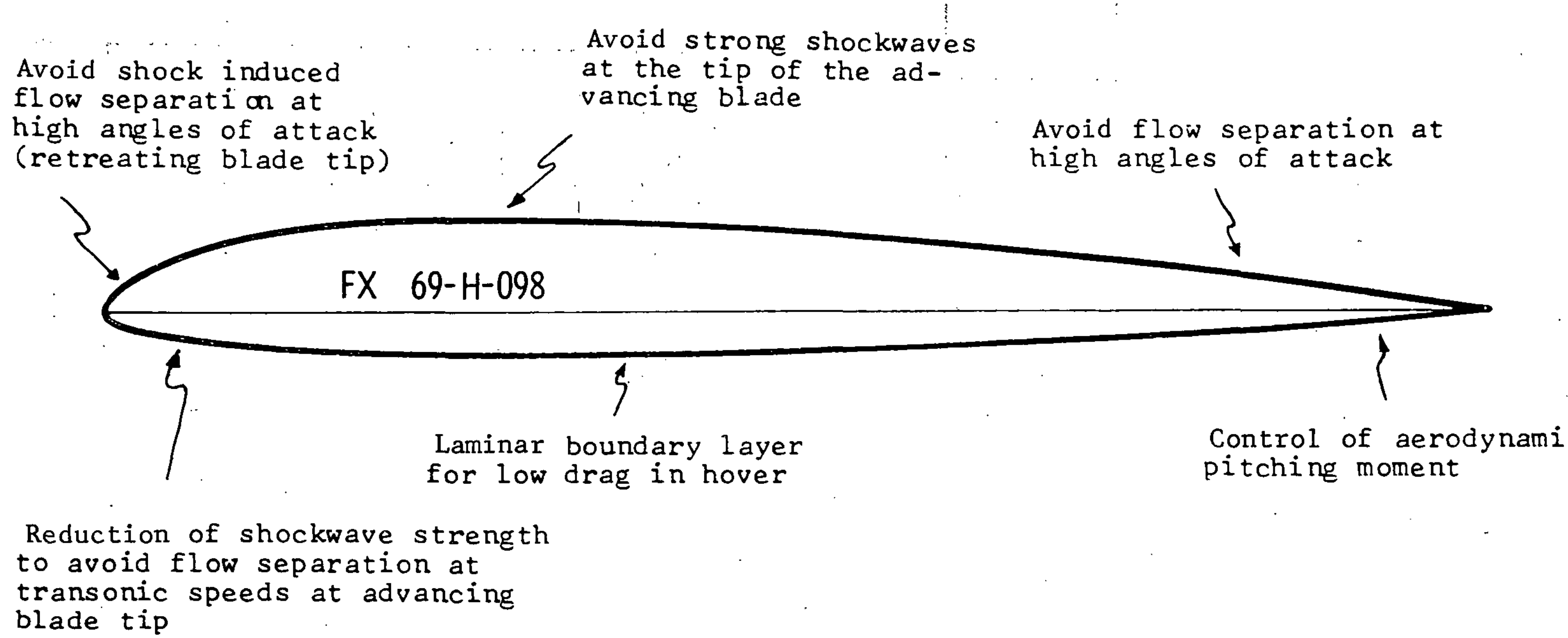


Figure 1. FX69-H-098 Airfoil and Design Considerations Used in Shaping Parts of the Contour

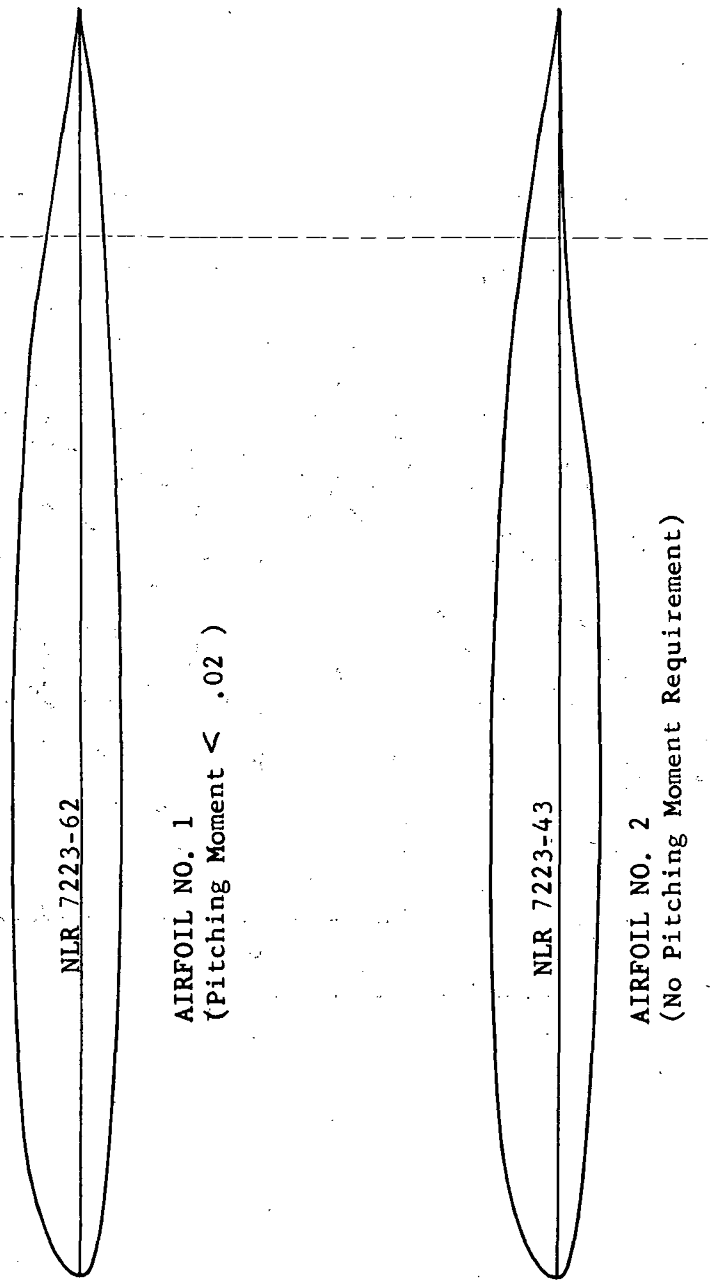


Figure 2. Final NLR Airfoil Sections

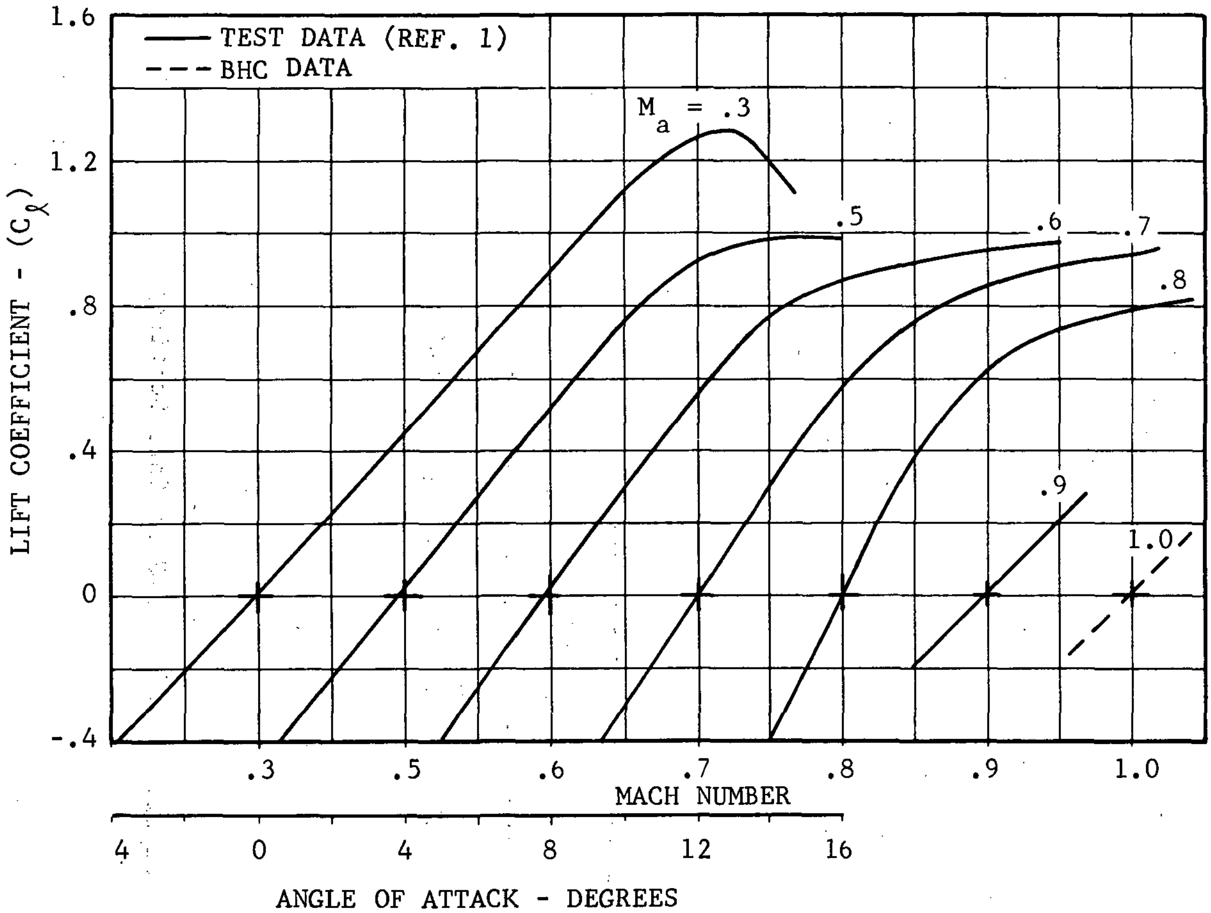


Figure 3. NACA 0012 Aerodynamic Section Data, Lift Coefficient Versus Angle of Attack

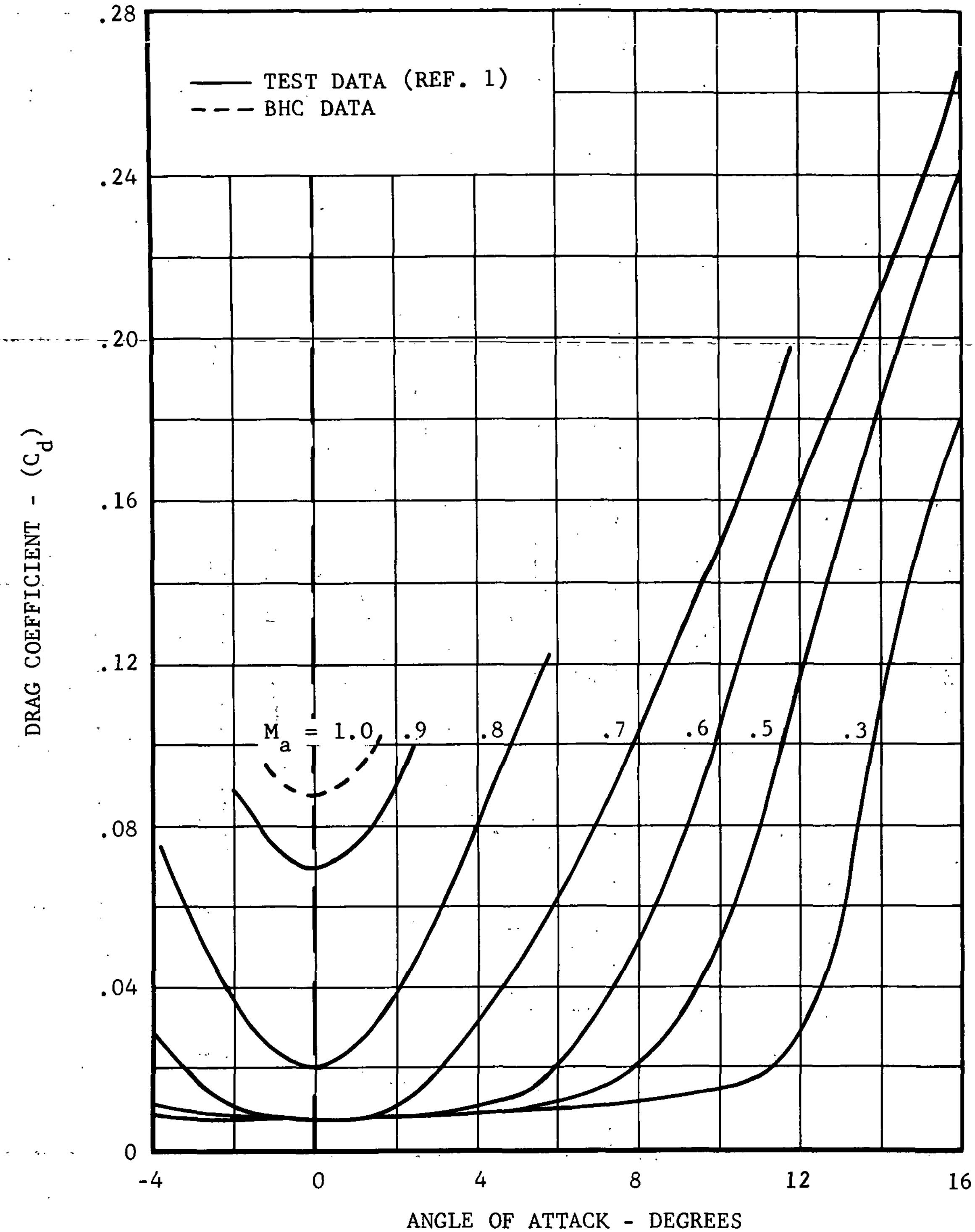


Figure 4. NACA 0012 Aerodynamic Section Data, Drag Coefficient Versus Angle of Attack

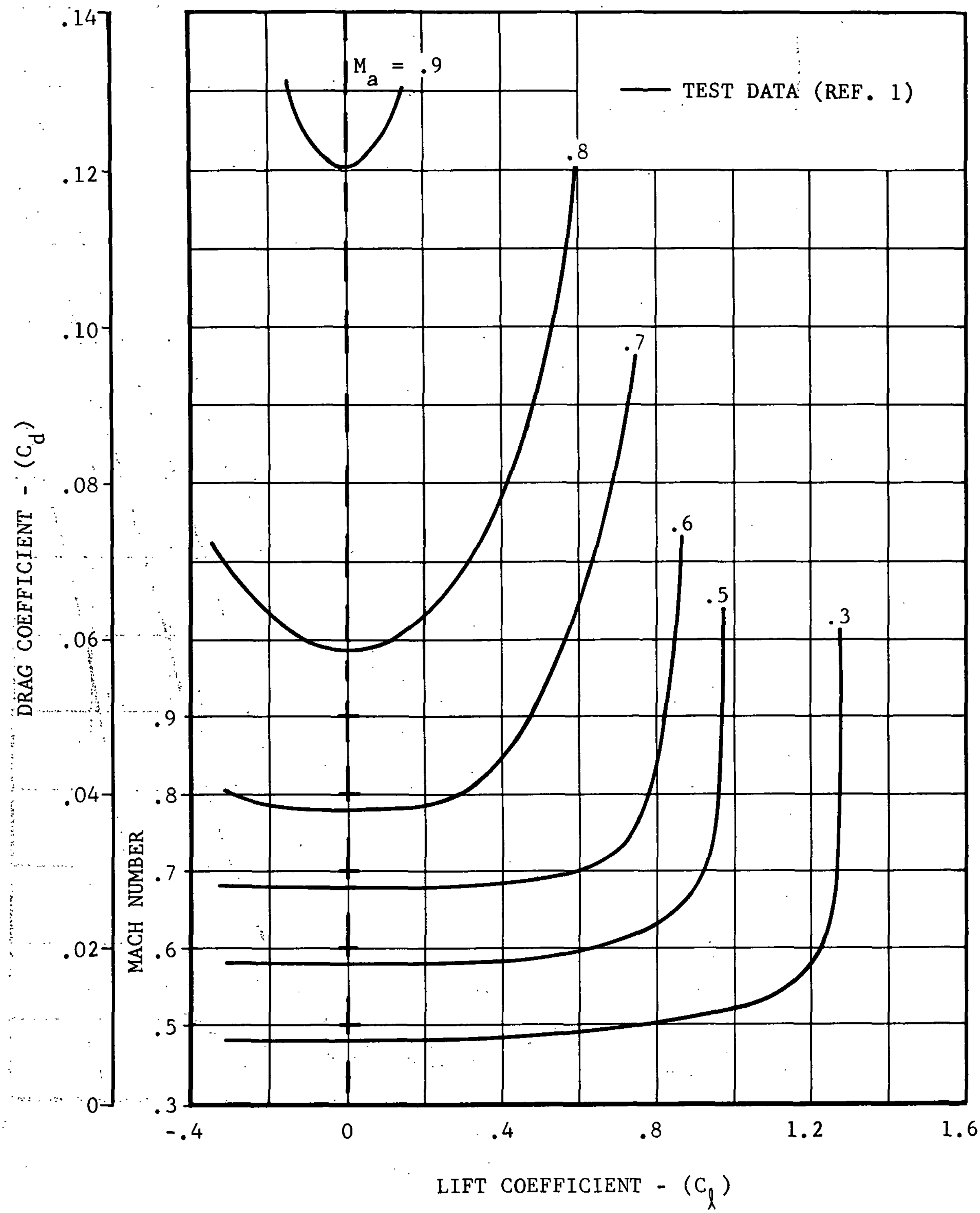


Figure 5. NACA 0012 Aerodynamic Section Data, Drag Coefficient Versus Lift Coefficient

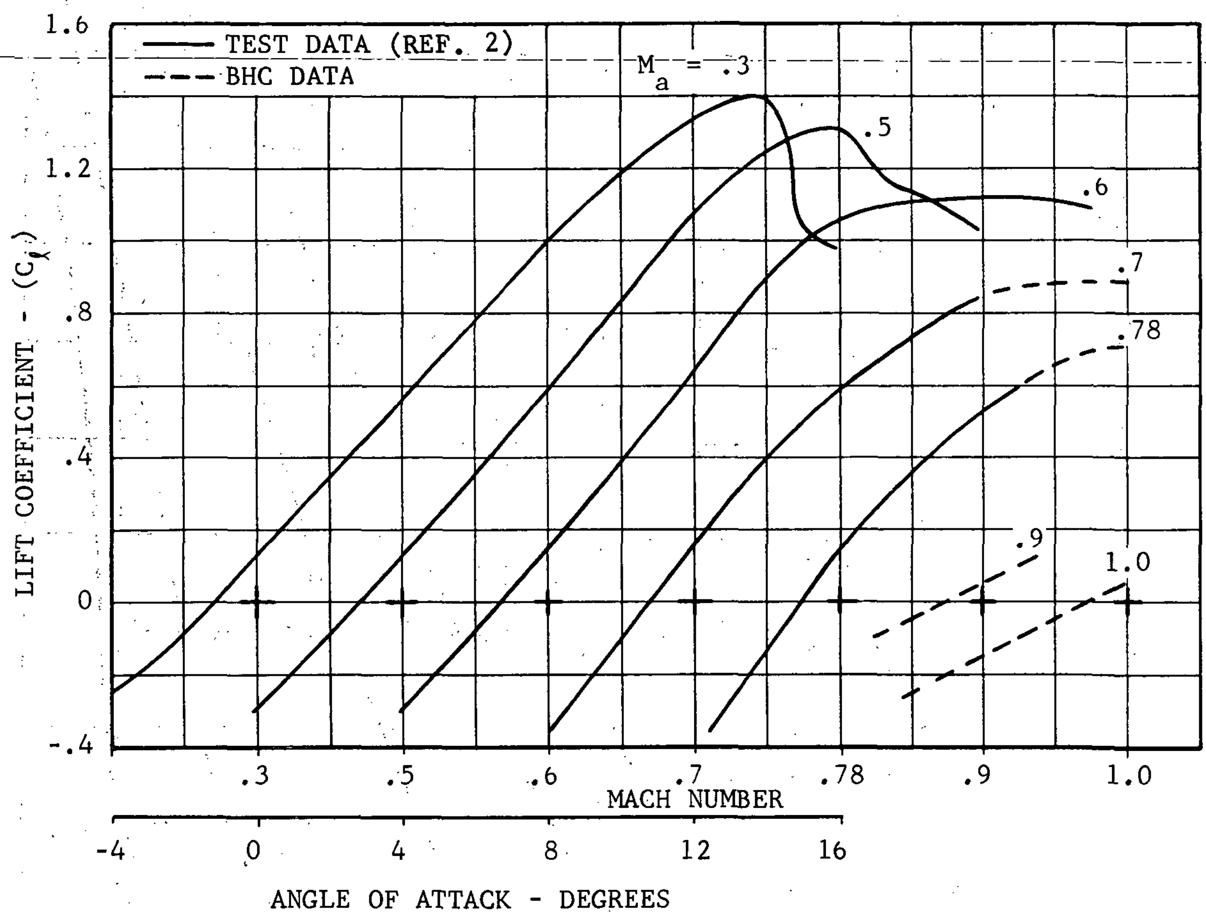


Figure 6. FX69-H-098 Aerodynamic Section Data, Lift Coefficient Versus Angle of Attack

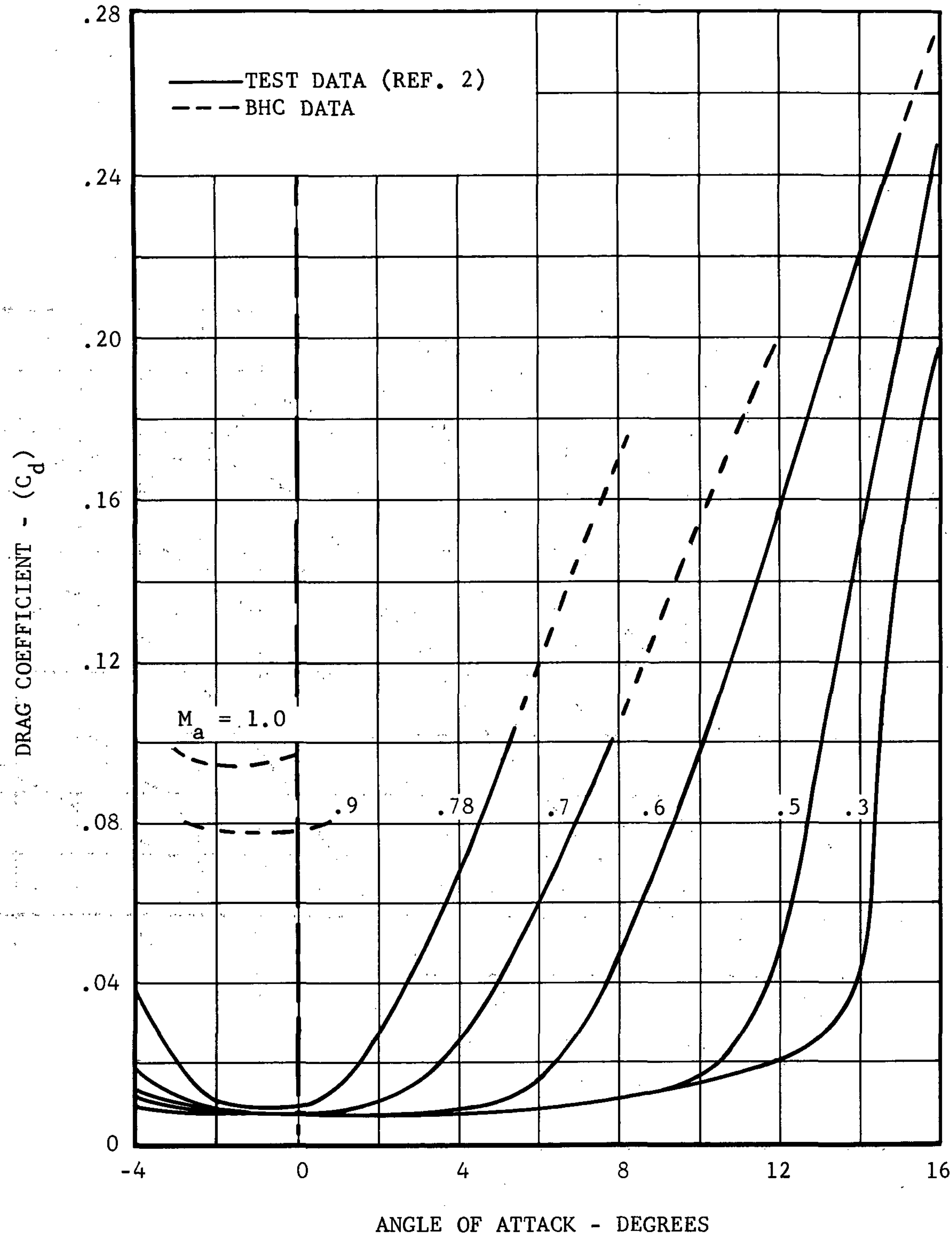


Figure 7. FX69-H-098 Aerodynamic Section Data, Drag Coefficient Versus Angle of Attack

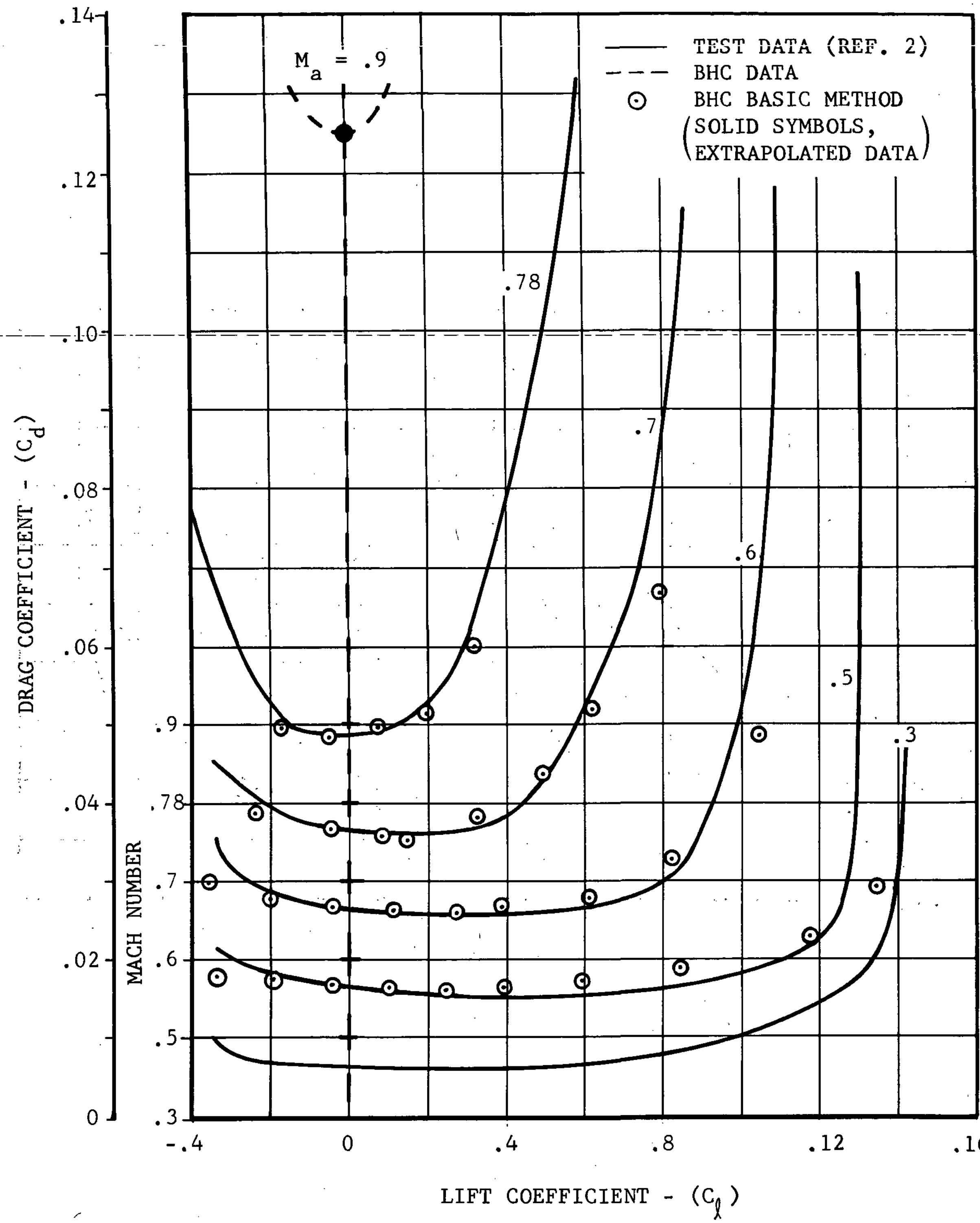


Figure 8. FX69-H-098 Aerodynamic Section Data, Drag Coefficient Versus Lift Coefficient

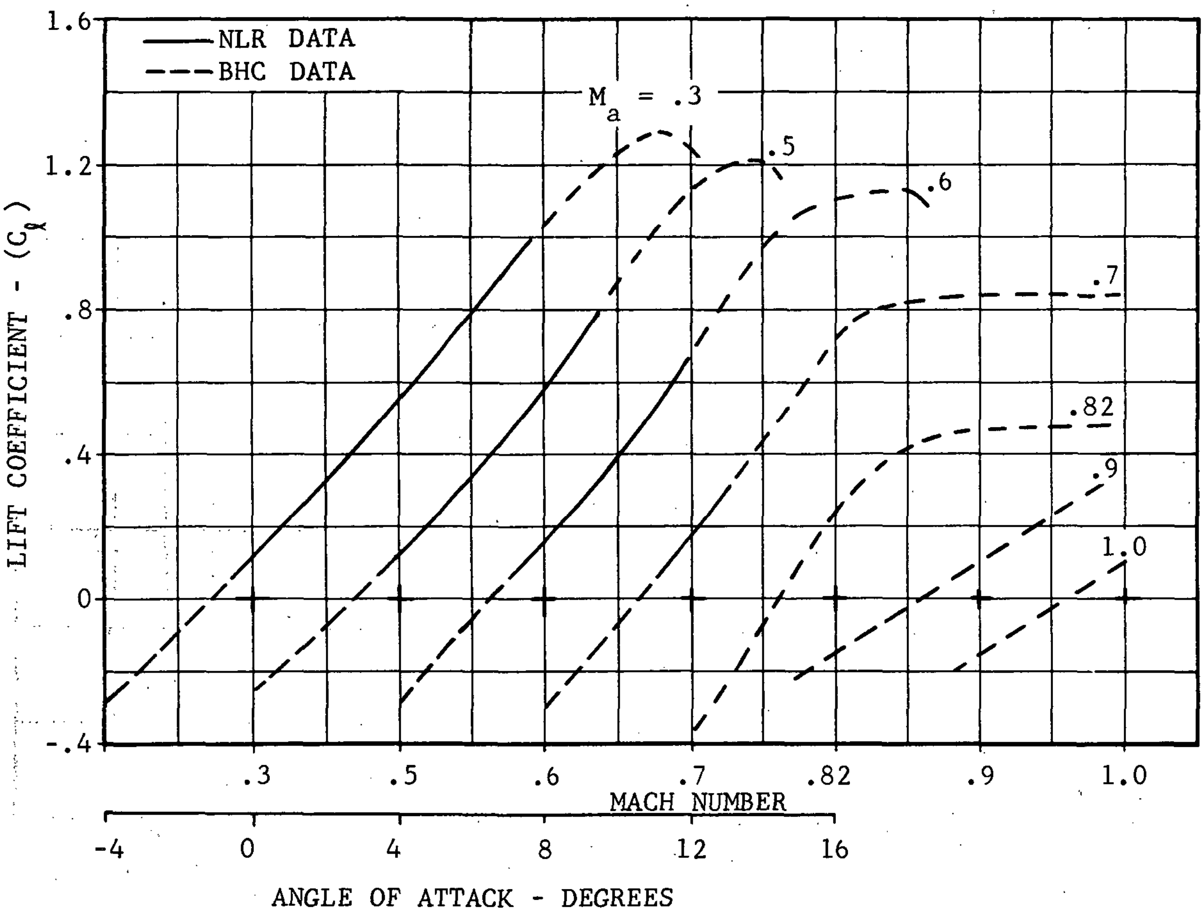


Figure 9. NLR 7223-62 (Airfoil 1) Aerodynamic Section Data, Lift Coefficient Versus Angle of Attack

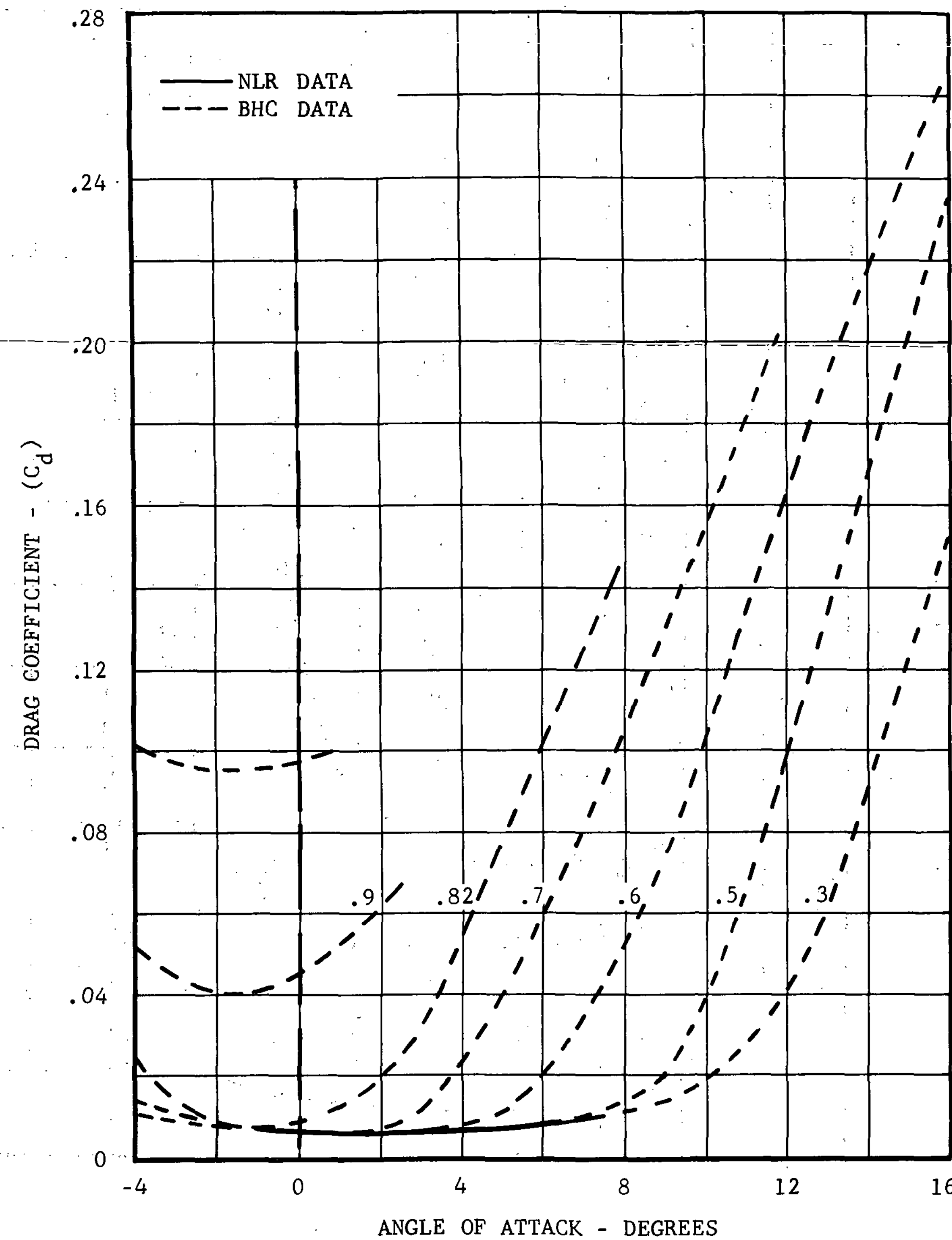


Figure 10. NLR 7223-62 (Airfoil 1) Aerodynamic Section Data, Drag Coefficient Versus Angle of Attack

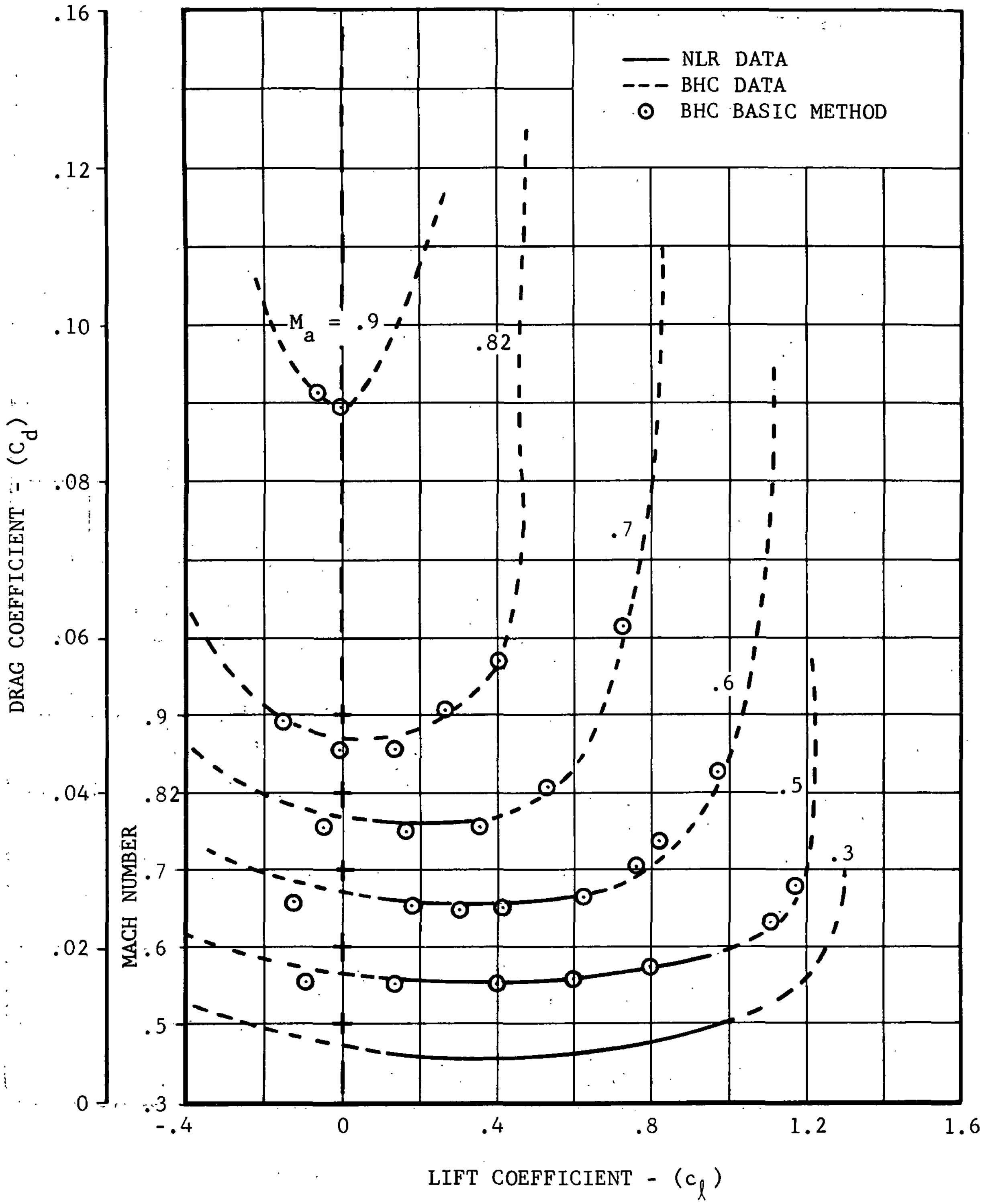


Figure 11. NLR 7223-62 (Airfoil 1) Aerodynamic Section Data, Drag Coefficient Versus Lift Coefficient

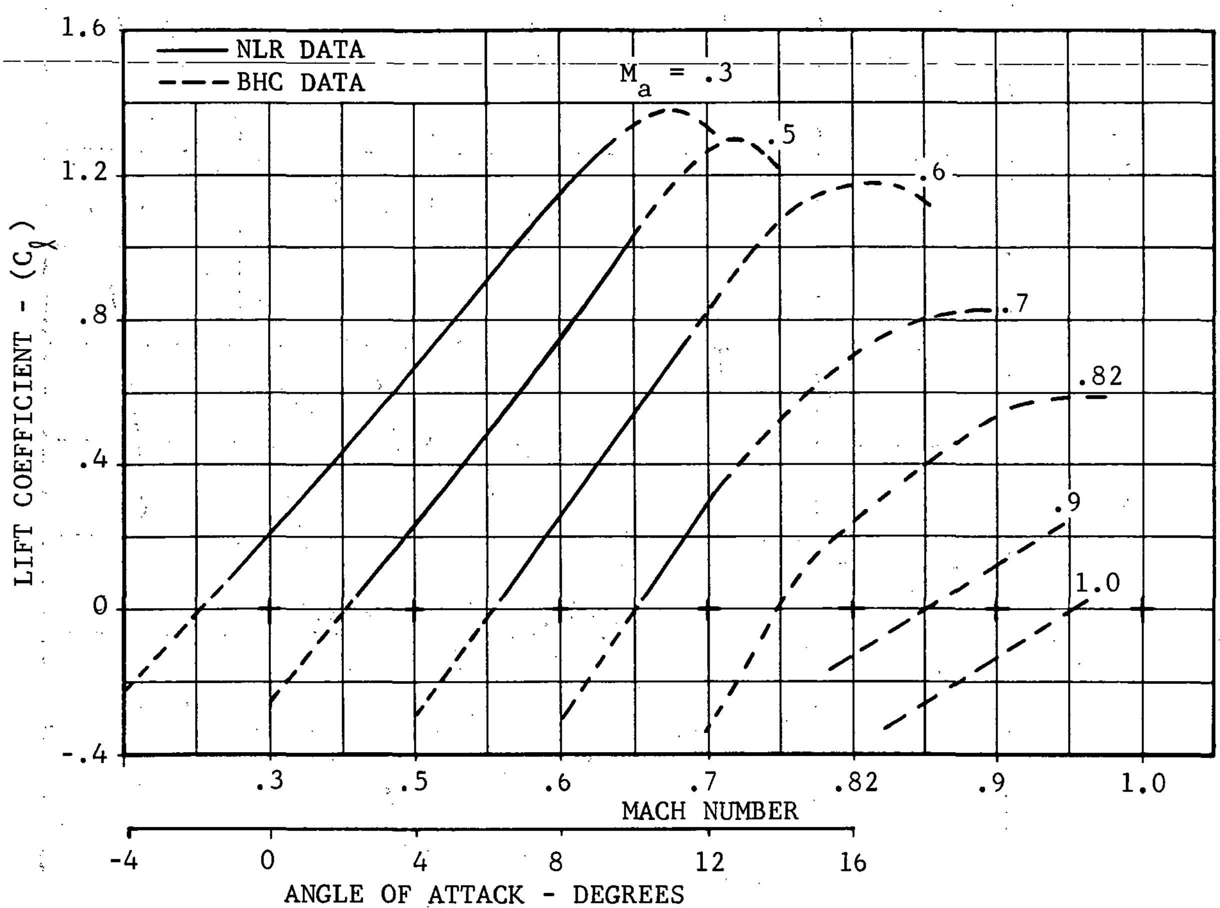


Figure 12. NLR 7223-43 (Airfoil 2) Aerodynamic Section Data, Lift Coefficient Versus Angle of Attack

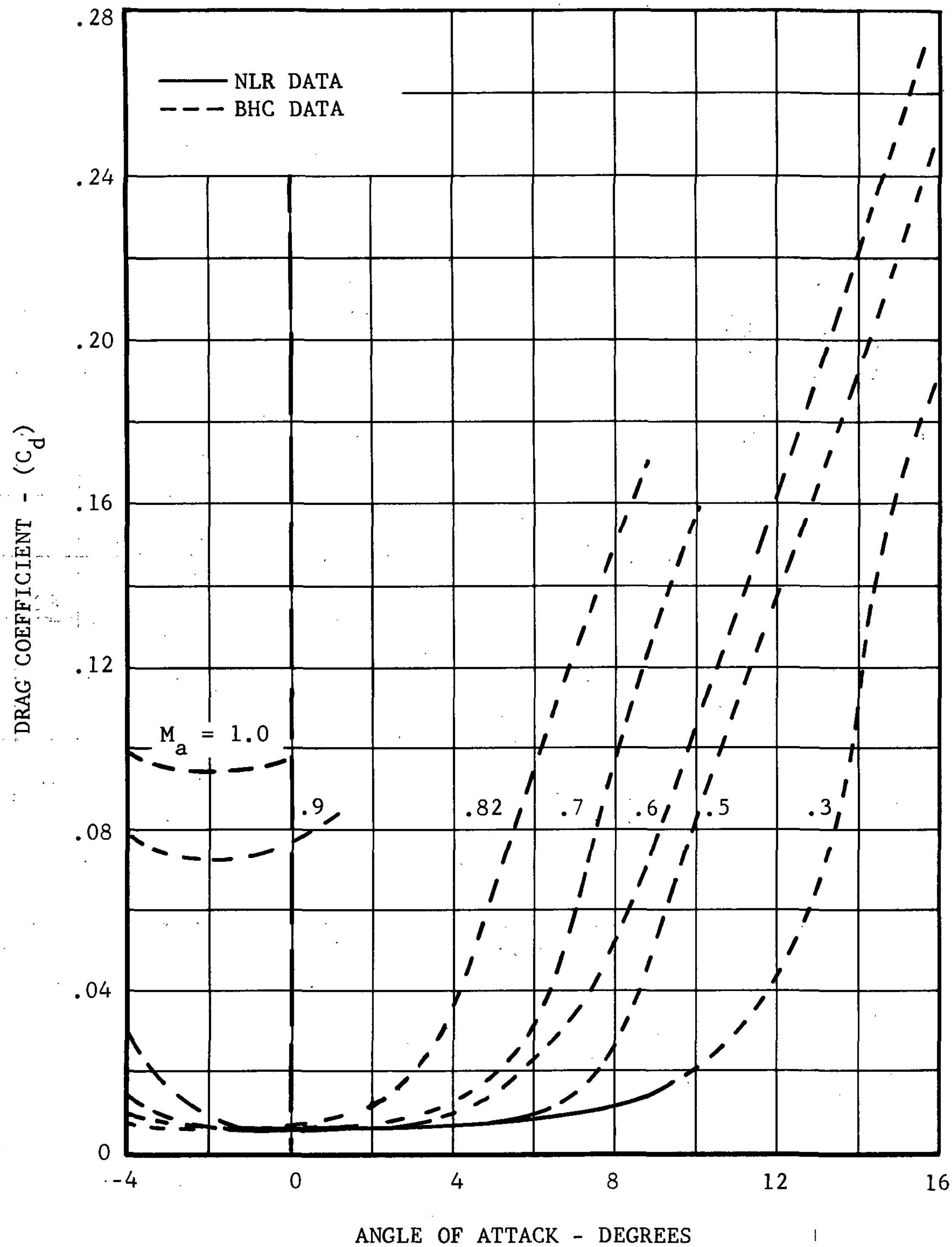


Figure 13. NLR 7223-43 (Airfoil 2) Aerodynamic Section Data, Drag Coefficient Versus Angle of Attack

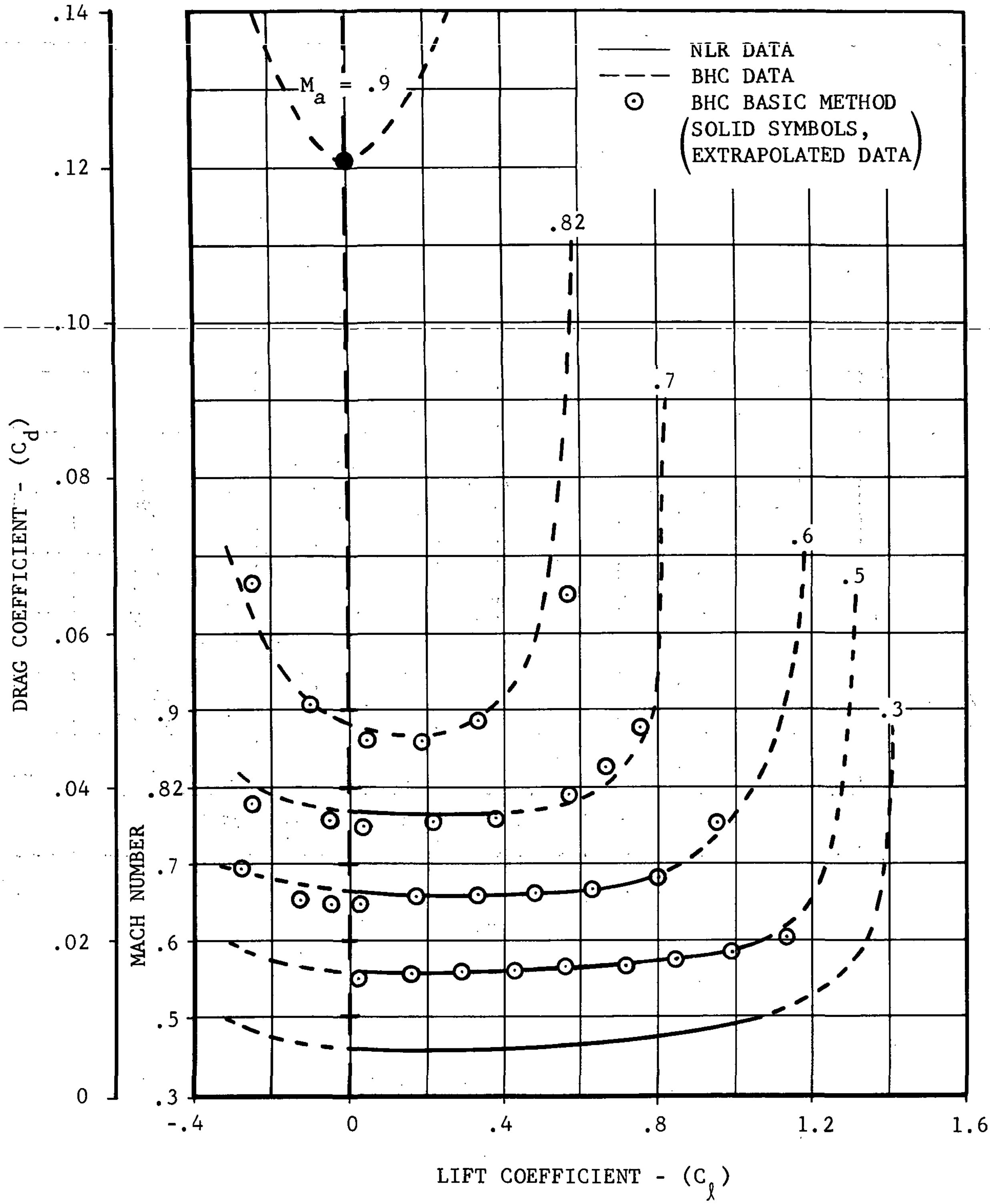


Figure 14. NLR 7223-43 (Airfoil 2) Aerodynamic Section Data, Drag Coefficient Versus Lift Coefficient

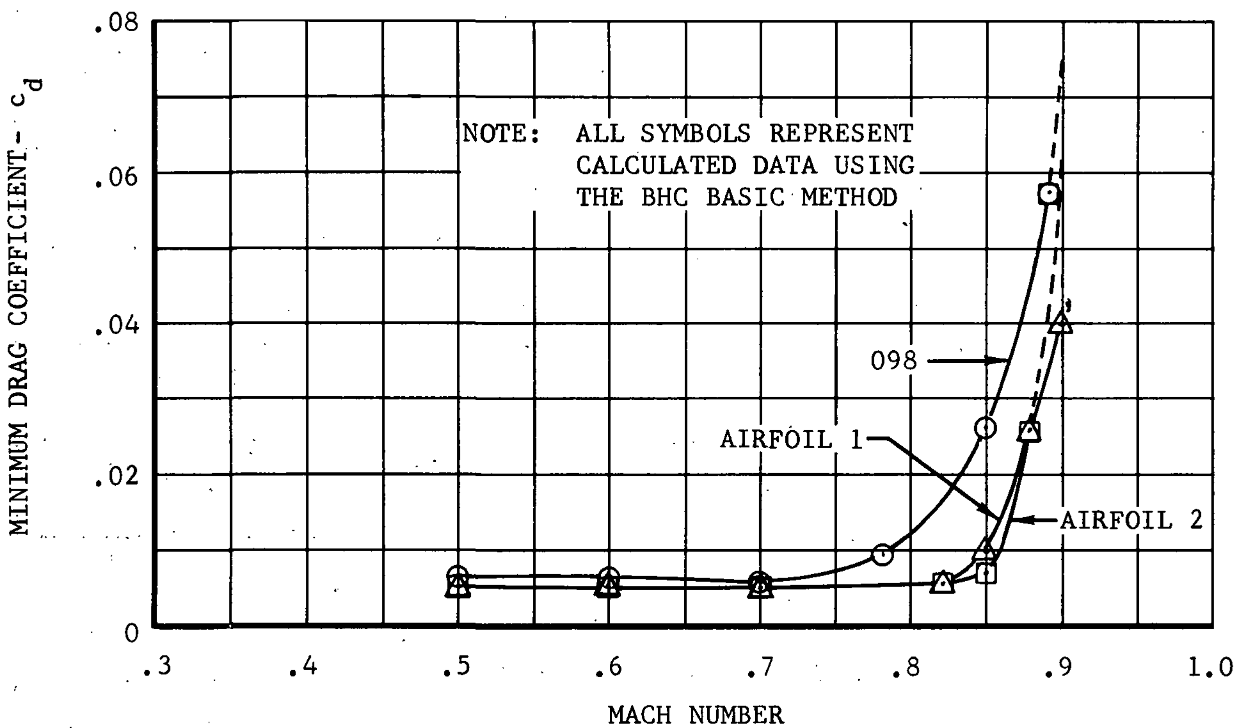


Figure 15. Section Minimum Drag Coefficient Versus Mach Number

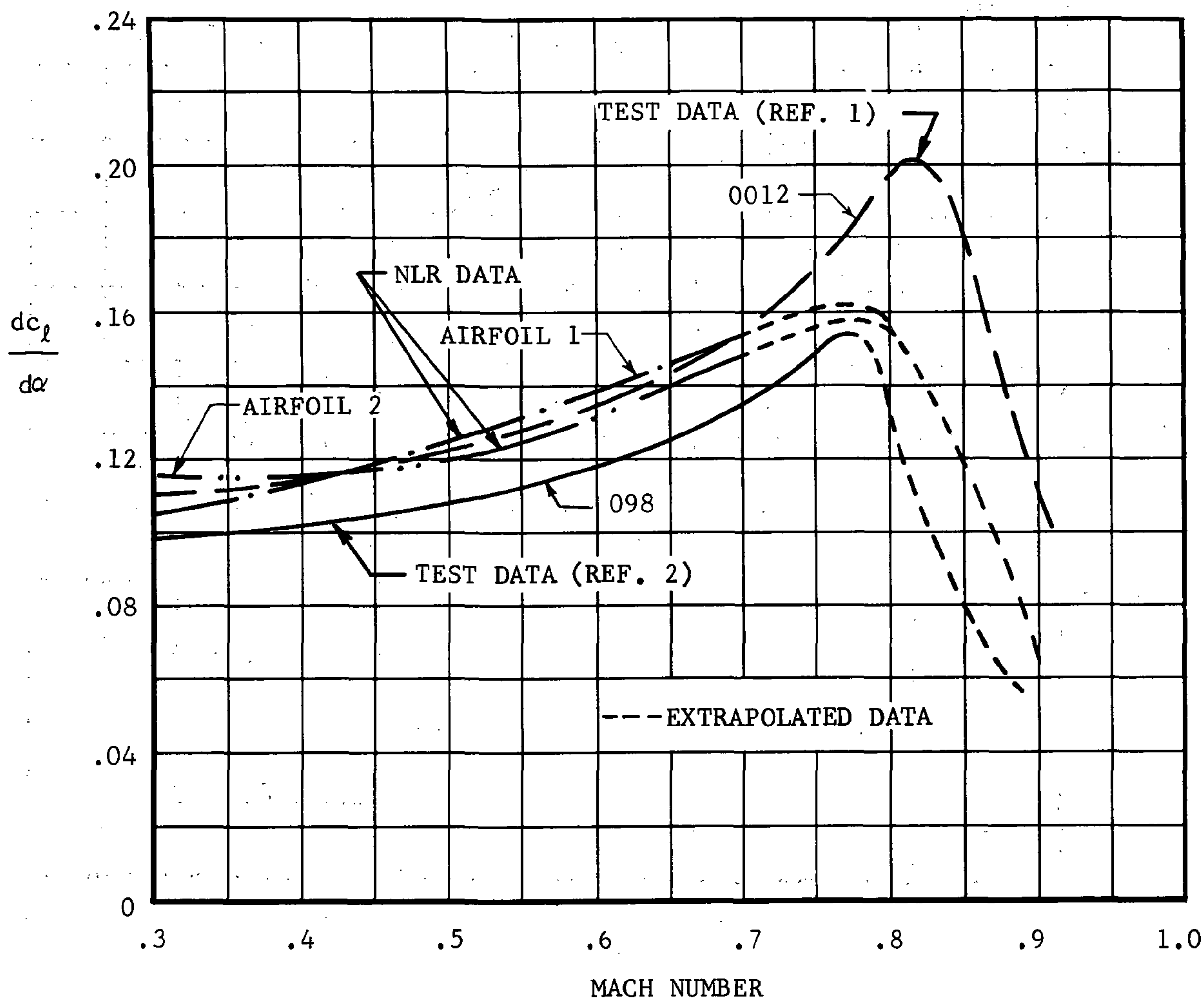


Figure 16. Section Lift Curve Slope Versus Mach Number

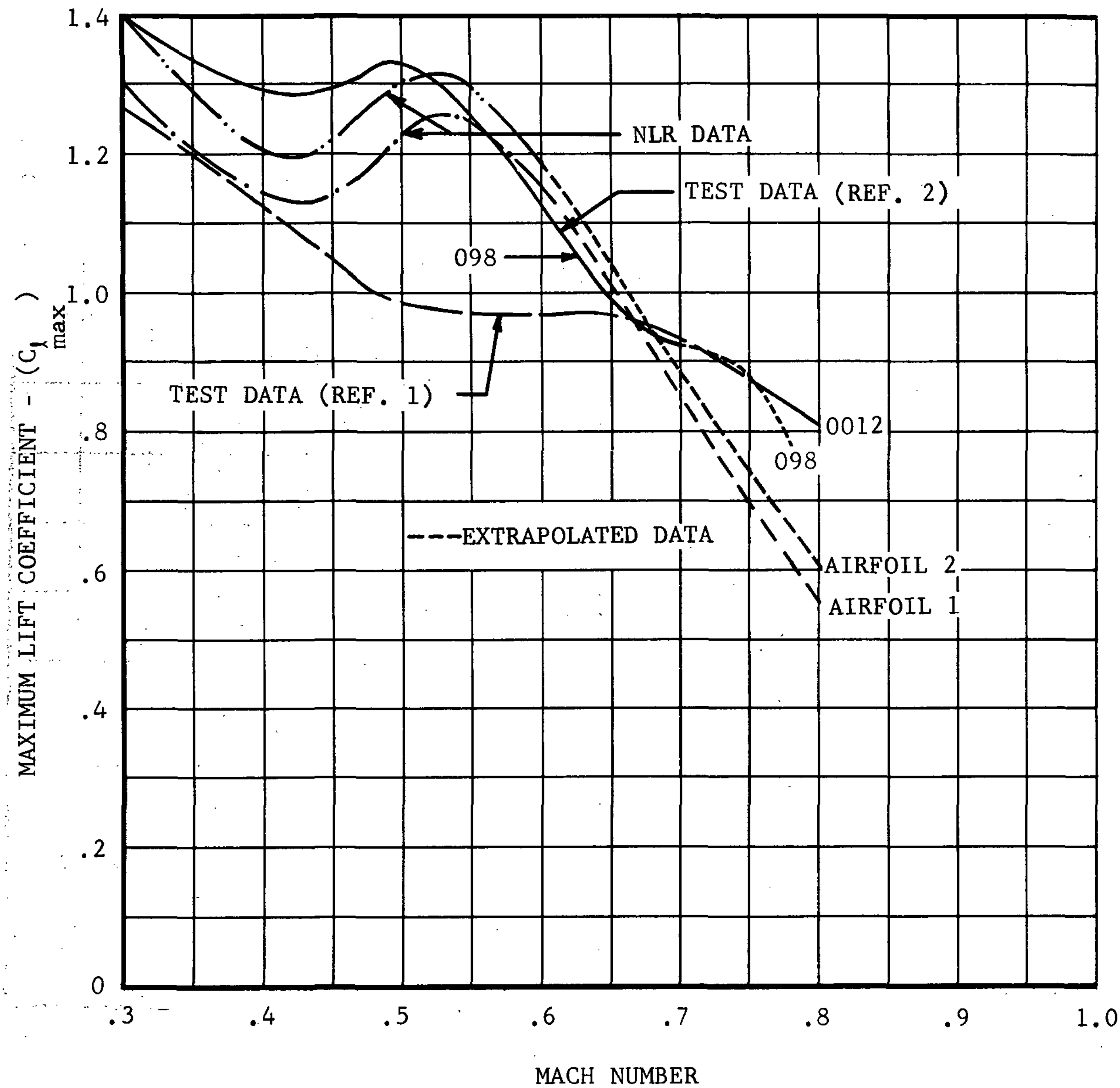


Figure 17. Section Maximum Lift Coefficient Versus Mach Number

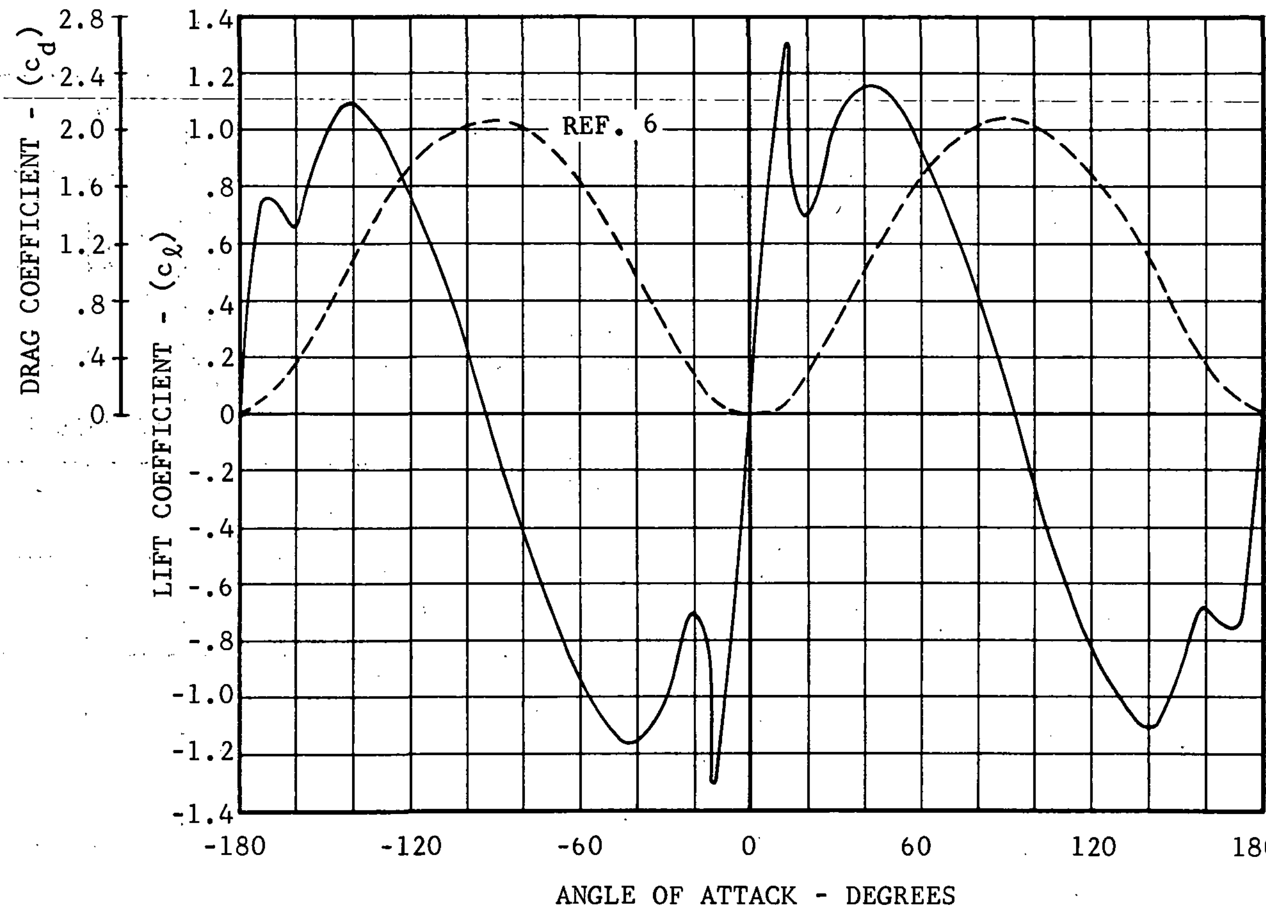


Figure 18. NACA 0012 Section Data For Large Angles of Attack

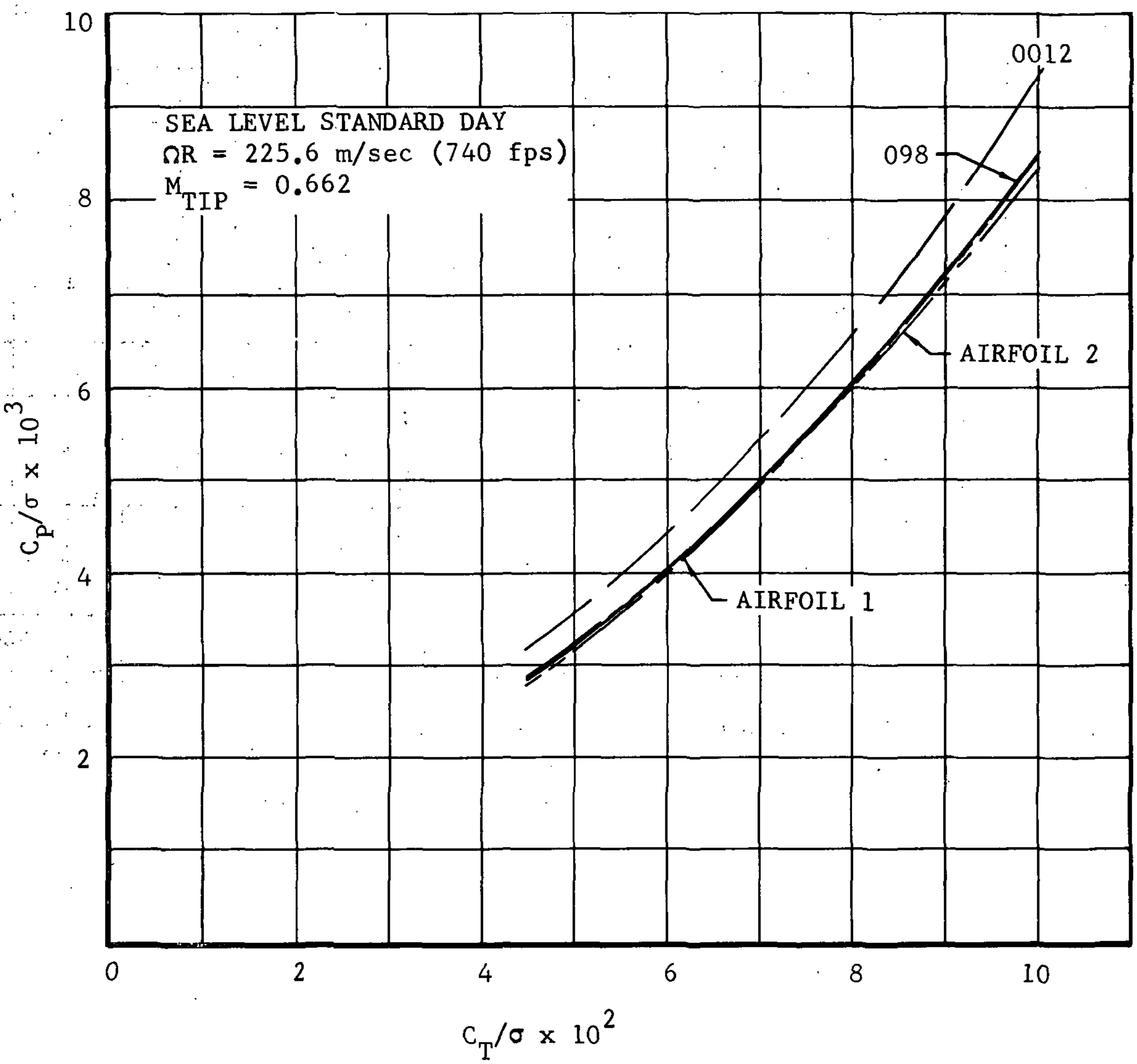


Figure 19. Hover Performance, 226 Meters Per Second (740 Feet Per Second) Tip Speed

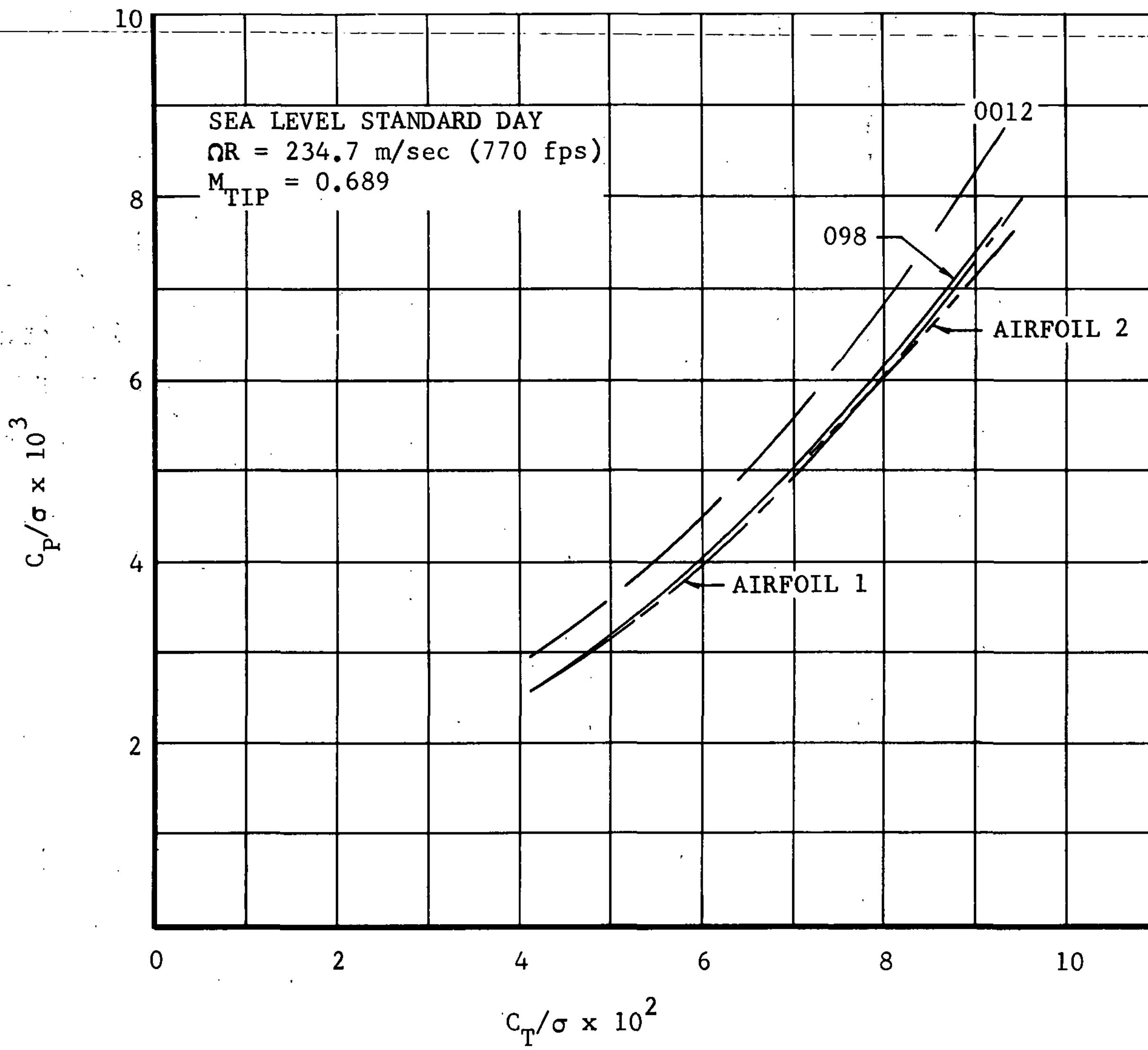


Figure 20. Hover Performance, 235 Meters Per Second (770 Feet Per Second) Tip Speed

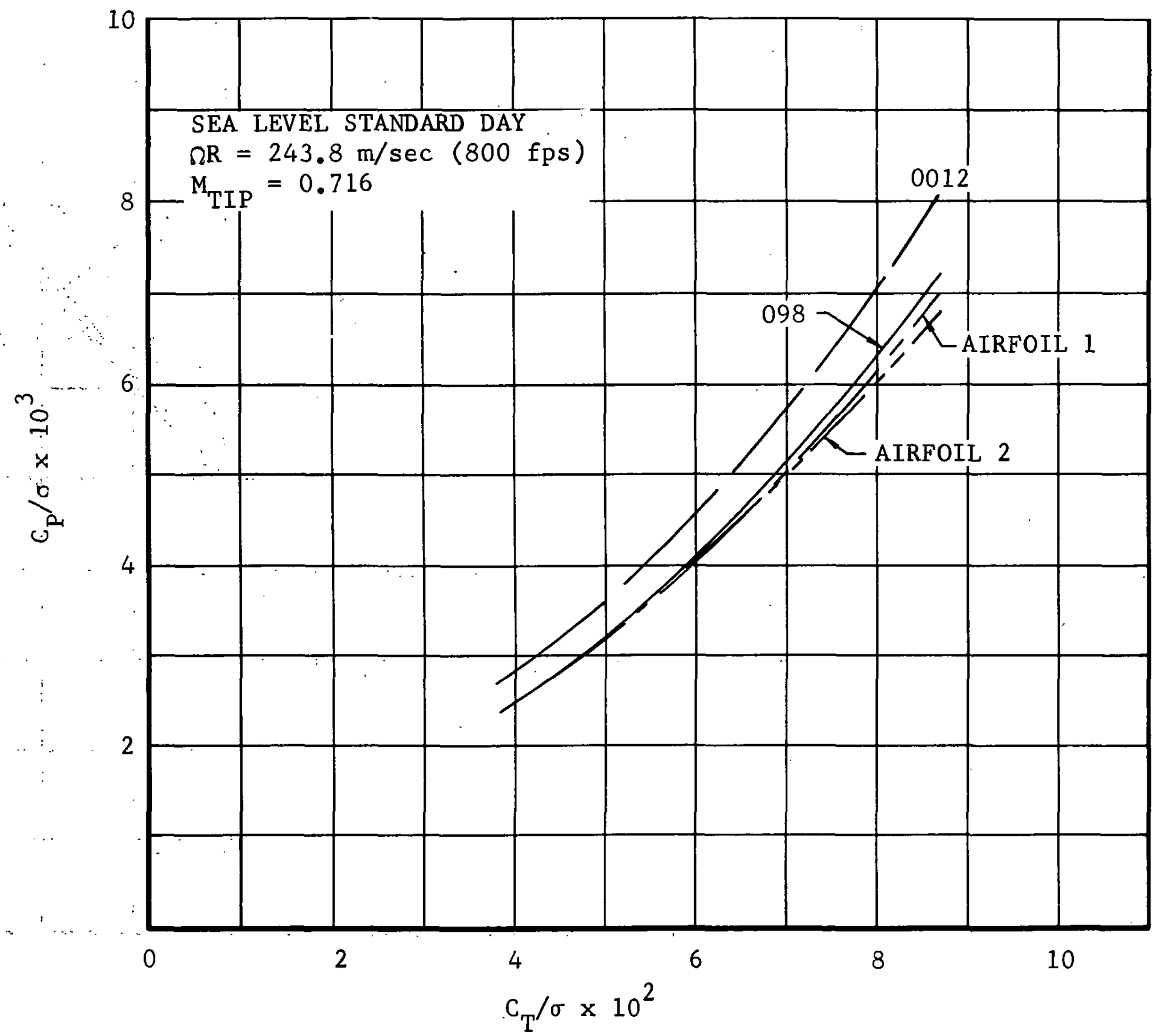


Figure 21. Hover Performance, 244 Meters Per Second (800 Feet Per Second) Tip Speed

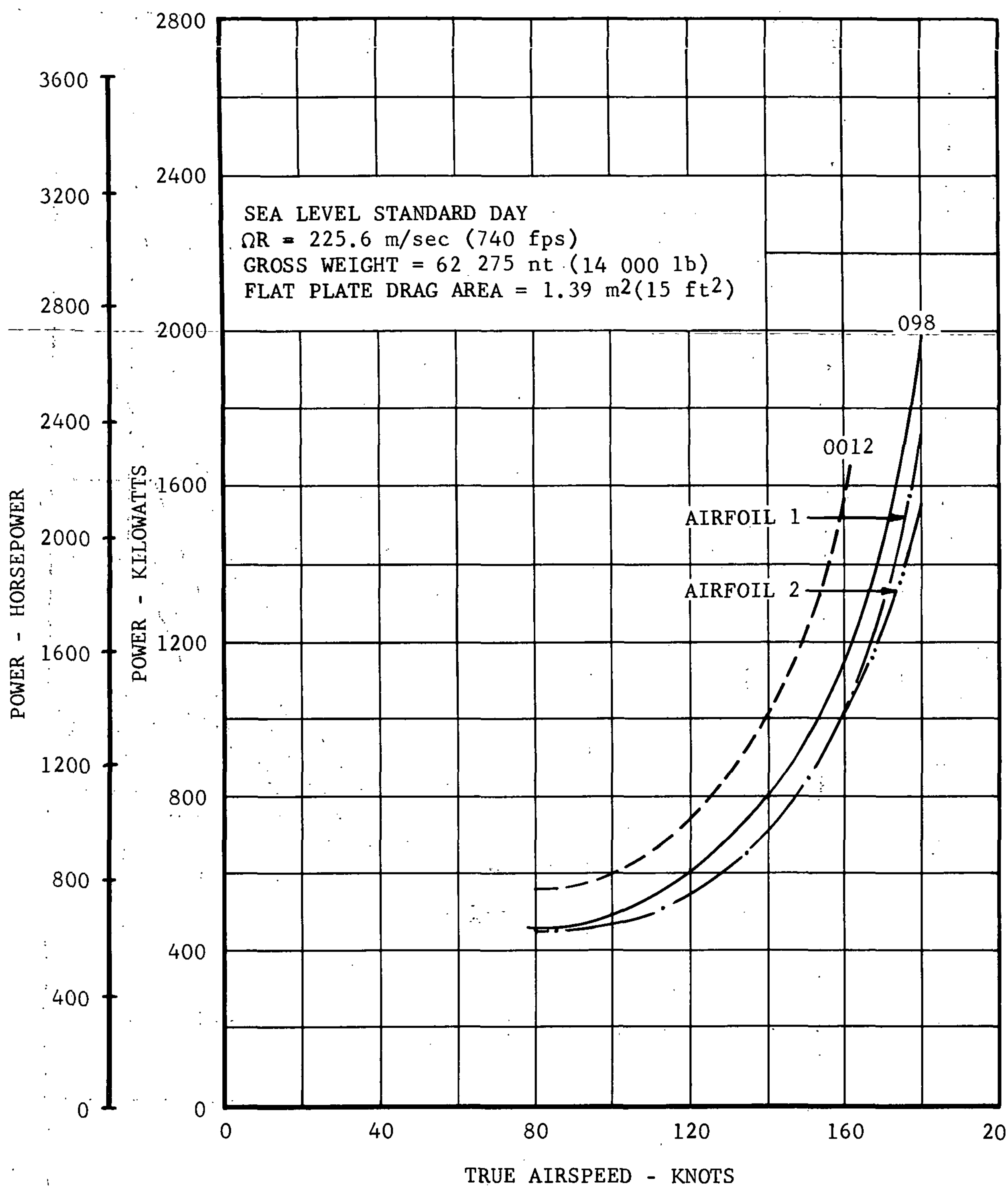


Figure 22. High Speed Forward Flight Performance, 226 Meters Per Second (740 Feet Per Second) Tip Speed

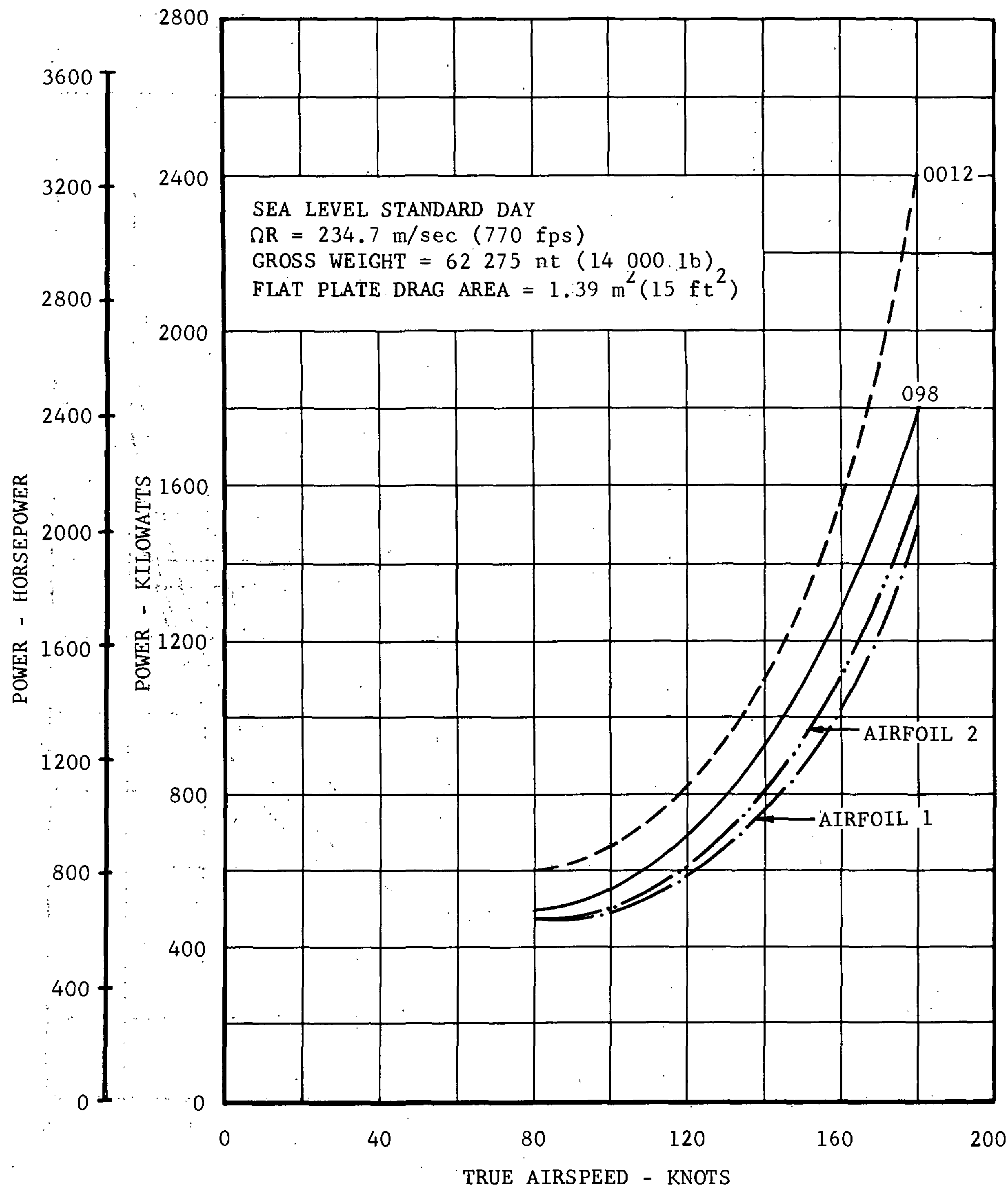


Figure 23. High Speed Forward Flight Performance, 235 Meters Per Second (770 Feet Per Second) Tip Speed

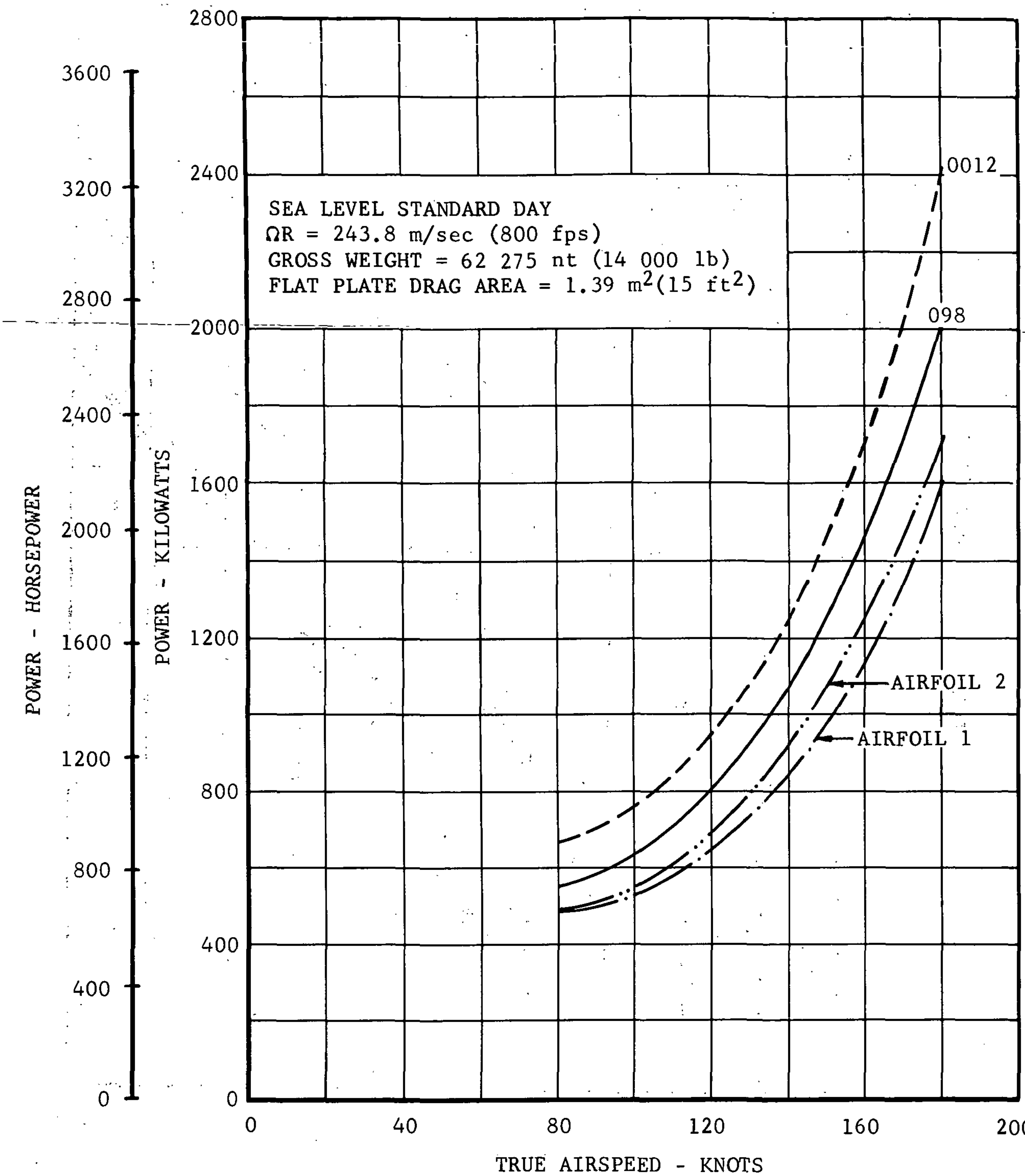


Figure 24. High Speed Forward Flight Performance, 244 Meters Per Second (800 Feet Per Second) Tip Speed

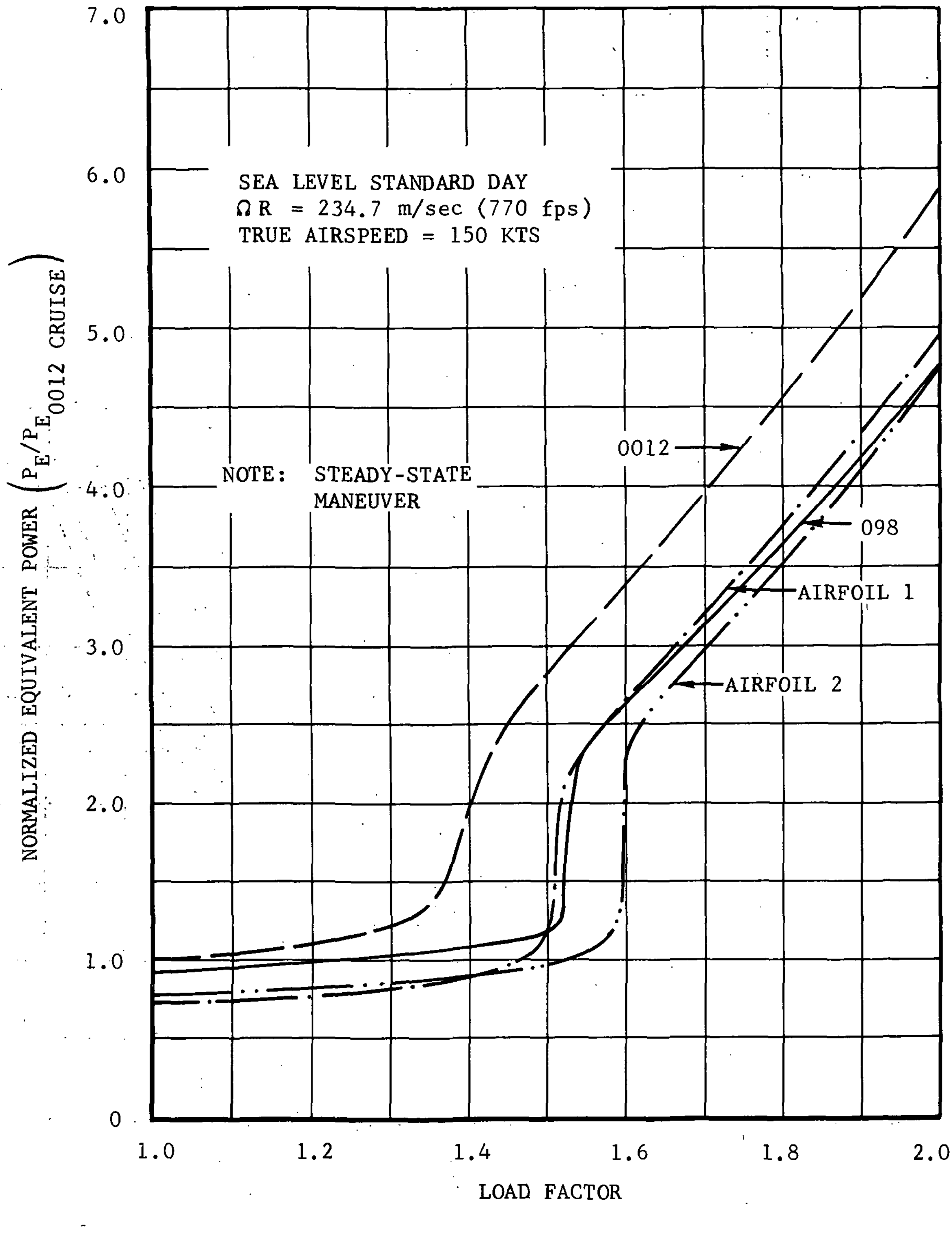


Figure 25. Maneuver Performance, Normalized Equivalent Power versus Load Factor

APPENDIX A

NATIONAAL LUCHT- EN RUIMTEVAARTLABORATORIUM

NATIONAL AEROSPACE LABORATORY NLR

THE NETHERLANDS

NLR TR 72140 C

AN AERODYNAMIC DESIGN STUDY FOR ROTOR AIRFOILS

by

J.W. Boerstoel *)
J.W. Slooff **)
G.H. Huizing +)
W.J. Piers ++)

SUMMARY

Using, as a basis, the hodograph method for transonic shockfree flow, a design study has been performed for airfoils satisfying the multiple design requirements that are typical for the helicopter rotor environment.

Two new airfoils have emerged from this study. The aerodynamic data predicted for the two airfoils compare well with contemporary rotor airfoils designed along more empirical lines, especially at high speed.

Of the two airfoils one was required to have a small pitching moment, the other not. It appears that the main implication of the pitching moment restriction is a reduction of $c_{\ell_{max}}$.

Both airfoils exhibit a characteristic peak in upper surface curvature that is believed to be essential for combining favourable high speed and manoeuvre performance. In this respect the possibilities of the hodograph method could not be fully explored. Continued parameter studies could possibly lead to further improvement.

*) Senior Mathematician
**) Senior Aerodynamicist
+) Senior Systems Analysist
++) Aerodynamicist

CONTENTS

Appendix/ 2 pages
4 tables
47 figures

LIST OF SYMBOLS

C_p	pressure coefficient $\frac{p_s - p_\infty}{q_\infty}$
c	chord length
c_d	drag coefficient
c_f	local skin friction coefficient
c_ℓ	lift coefficient
c_m	quarter chord pitching moment coefficient
L/D	lift/drag ratio
Ma	Mach number
p_s	local static pressure
p_∞	free-stream static pressure
q_∞	free-stream dynamic pressure
R	radius of surface curvature
Re	Reynolds number based on chord length
t	airfoil thickness
x,z	airfoil system of coordinates (x chordwise)
α	angle of attack
ϵ_o	thickness parameter
Γ	circulation
λ_1	parameter controlling nose bluntness
λ_2	parameter controlling camber
θ	surface slope

SUBSCRIPTS

ℓ	refers to lower surface
max	maximum value
min	minimum value
o	zero lift quantity
t.e.	refers to trailing edge
u	refers to upper surface

SUPERSCRIPTS

*	critical value for local sonic flow

1 INTRODUCTION

In recent years a growing interest can be noticed in the development of "advanced" airfoils specifically designed for helicopter rotors (Refs.1,2,3). The reasons are twofold.

First, there is the evolutionary type of expansion of the helicopter capabilities with regard to efficiency and performance which leads to a need for aerodynamically more efficient rotors. The flow through a helicopter rotor is of a complex, three-dimensional and unsteady nature. However, it has been verified at several occasions that the performance of a rotor depends strongly on the two-dimensional steady characteristics of the rotor profile. Although the detailed nature of this dependence is not clear, the performance of a helicopter rotor in a given flight condition can be improved by improving the characteristics of the rotor airfoil section in the two-dimensional, steady flow conditions.

Secondly, new analytical design tools and concepts are currently becoming available.*) This makes it possible to deal more adequately with the transonic effects that occur in most of the helicopter's flight conditions. Such developments, to a certain extent, parallel the recent work on supercritical airfoils for fixed-wing airplanes.

The initial stimulus for developing airfoils with favourable transonic characteristics was given by Pearcey (Ref.5), who proved experimentally, that shock-free transonic flow is a real possibility. He found, that shockwaves can be reduced in strength and even eliminated by designing for a "peaky" type of pressure distribution.

With respect to rotor airfoil design a considerable step forward was made through the work of Wortmann (Ref.1), who applied the "peaky" principle to improve the transonic characteristics. In particular, he showed how to utilize a "peaky" pressure distribution to increase the maximum lift in the medium Mach number range of the retreating rotor blade.

While Pearcy had set out empirical rules for the design of shock-free airfoils, a mathematical solution to the problem was

*) For a recent survey article see reference 4.

given by Nieuwland (Ref.6). Using analytic hodograph theory Nieuwland and his collaborators calculated a family of shock-free profiles of considerable geometrical variety classified as quasi-elliptical airfoils. Results of two-dimensional tests in a transonic tunnel verified the theory for both non-lifting and lifting airfoils (Refs.7,8).

A significant contribution to the understanding of the physics of transonic shock-free flow was given by Spee (Ref.9), who showed that the flow around quasi-elliptical airfoils is stable with respect to unsteady disturbances. He found that the shock-free design condition is embedded in an interval of free stream Mach numbers and angles of attack where the wave drag is negligeable: The design condition can be reached in a continuous stable manner from neighbouring conditions.

Since then, it has been found, that the definition of 'neighbouring conditions' can be extended to include certain contour deviations from the analytical airfoil shape. I.e., (subsonic) parts of an analytic shape can be modified without destroying the low-drag properties of the basic shock-free flow. This provides an additional degree of freedom that can, for instance, be used to increase the "shock-free lift coefficient" at a given Mach number (Ref.10).

This report describes an effort to make profitable use of the hodograph method for quasi-elliptical airfoils in a design process for helicopter rotor airfoils. The motivation for this investigation emerged from joint deliberation between Bell Helicopter Company (BHC) and NLR. It was expected that designing for shock-free flow by means of the analytic hodograph theory, rather than for "peaky" pressure distributions as Wortmann (Ref.1) did, would lead to a further improvement of performance.

The work forms part of a NASA/BHC contract and was executed by NLR under subcontract to BHC. In the proposed program (Ref.11) BHC was feeding-in the helicopter experience, the design requirements and a baseline helicopter airfoil section designed by Wortmann. Based on these data NLR was to design two airfoils ; one that satisfies the usual low pitching moment requirement for rotor airfoils and one that does not. Comparison of the two airfoils

would provide an answer to the question whether the pitching moment restriction causes a degradation in aerodynamic performance.

In the next chapter we will first discuss the design requirements provided by BHC. This will be followed by a description of the design procedure, a discussion on the computations with the hodograph method and a description of the process of modification of the basic shock-free shape selected. The final chapters summarise and discuss the results obtained in terms of airfoil performance.

2 DESIGN REQUIREMENTS

As discussed in detail reference 1 the helicopter's usual flight conditions, distinguished as hovering, manoeuvering and high speed flight (Fig.1), imply conflicting requirements in airfoil design. The hover condition asks for a high lift/drag ratio at a lift coefficient of the order of 0.65 over most of the rotor blade span at Mach numbers upto approximately 0.6 for the outboard sections. In manoeuvre the maximum g-capability is directly related to the $c_{\ell_{max}}$ of the retreating blade, which, for the outboard sections, should be as high as possible in the 0.4 to 0.6 Mach number range. In high speed flight the advancing blade tip requires a high drag rise Mach number at low values of c_ℓ.

Conflicting as the rotor airfoil requirements are, there is little sense in pursuing one requirement first and getting concerned about the others later. Rotor airfoil design is an art of compromise from the outset and the tentative specifications set out for this design study do already represent some form of compromise. This may be illustrated by comparing the tentative specifications for this design study with the characteristics that (see appendix) could possibly be obtained if each of the requirements is optimized for separately (table 1).

Comparing the tentative design figures with those realised experimentally for a contemporary rotor airfoil (Wortmann FX69-H-098 airfoil, reference 15) it can be seen that in the present design study attention is focussed on improving the high speed characteristics without, however, affecting the performance at the other design points. Both the Wortmann and the tentative figures represent

a substantial improvement with respect to the "standard helicopter airfoil", NACA 0012.

The procedure followed in trying to meet the design specifications is described in the next chapter.

3 DESIGN CONSIDERATIONS AND DESIGN PROCEDURE

3.1 Characteristics of the baseline airfoil (Wortmann FX69-H-098)

Prior to going into the details of the procedure followed in the present design study, it is worthwhile to consider the characteristics of a typical rotor airfoil. The Wortmann FX69-H-098 was chosen for this purpose because it combines the multiple rotor airfoil requirements in a highly successful way. The considerations used in shaping this airfoil are summarized in figure 2, which was taken from reference 11.

In terms of aerodynamic characteristics these considerations have, at the hover condition, resulted in laminar, accelerating flow over almost the entire lower surface. On the upper surface there is a small supersonic zone between 6 % and 16 % chord, terminated by a weak shock wave (Fig.3). At manoeuvre it exhibits a "peaky" type supersonic region, extending from the leading edge to 10 % chord, that is terminated by a strong shock, leading to shock-induced stall. At high speed there is a supersonic zone between 10 % and 60 % chord on the upper surface, terminated by a strong shock and a "peaky" pressure distribution with weaker shock(s) on the lower surface.

In terms of geometry the 10 % thick FX69-H-098 exhibits a fair amount of nose droop, a nose radius of 0.6 % chord and, in order to reduce the pitching moment, a slightly reflexed camber line at the trailing edge. Maximum thickness occurs at 30 %. A very important characteristic appears to be the upper surface curvature distribution at about 10 % chord ($\sqrt{x/c} \approx 0.33$ in figure 4). Wortmann (Refs.1,16) found this to be a key feature to a high $c_{\ell_{max}}$ at Ma=0.5. As will be discussed later, it has also some beneficial consequences for the high Mach number, low c_ℓ conditions.

3.2 Considerations underlying the approach selected

In view of the fact, that the tentative design specifications of table 1 do not differ extremely from those realized by the FX69-H-098 airfoil, it may be anticipated that the geometry and pressure distributions for the new airfoils must exhibit some similar features in order to meet the requirements. The question is how to make use of the hodograph method in order to obtain such features.

In order to appreciate the following line of thought it is essential to realise that the hodograph method for quasi-elliptical airfoils provides for given values of input parameters one and only one shock-free shape and corresponding pressure distribution as output. Of the input parameters one is related to the free stream Mach number and one to the circulation around the airfoil. The other parameters can be used to generate a certain family of shapes for one particular design point in the c_ℓ-Ma plane.

Leaving aside the hover condition, which appears not very critical with respect to transonic effects, this knowledge about the hodograph method suggests two alternative ways of approach. One is to calculate a suitable shock-free shape for a high c_ℓ at Mach 0.5. Using the freedom mentioned in chapter 1, to modify parts of such a basic airfoil one could optimise further towards hover and high speed. The other possibility is to start out from high speed and optimise towards hover and manoeuvre.

For several reasons it was decided to start out from the high speed side. In the first place it is clear from the extent of the supersonic flow regions in figure 3, that the high speed condition will determine a much larger part of the airfoil contour than the manoeuver $c_{\ell_{max}}$ requirement. This motivation may seem to contain a paradox at first sight, because the manoeuvre condition would give more freedom to modify the basic shape. However, shaping for transonic shock-free flow is a very delicate matter and one certainly would like to leave this to the hodograph method as much as possible.

A second reason is constituted by the fact that viscous effects, including shock-induced separation, play a dominant role at $c_{\ell_{max}}$. Although, one could in principle, design for a high, shock-free c_ℓ at Ma=0.5, there still would be considerable

uncertainty as to the eventual $c_{\ell_{max}}$. At high speed, low c_ℓ, where the Mach number for rapid drag rise is the quantity of importance, the situation is more predictable.

3.3 The actual design process

Having chosen the high speed approach we will now discuss the design procedure more specifically. For this purpose consider the flow diagram of figure 5. The diagram on the left hand side illustratesthe purpose of the different steps in the design procedure. The similar diagram on the right lists the analytical methods used in the different steps.

The first step is to calculate by means of the hodograph method (Refs.6,17) a series of shock-free shapes that will satisfy the high speed requirement. This step needs systematic variation of the input parameters involved. By engineering judgement of geometry and pressure distribution the shapes that are most promising are selected for further evaluation. This further evaluation consists of a crude estimate of the hover and manoeuvre characteristics. In the present investigation the approximate subsonic potential flow method of reference 18 was used for this purpose. The shape that promises the highest $c_{\ell_{max}}$ at Ma=0.5 is then chosen as the basic high speed airfoil.

The next step is to modify parts of the basic airfoil with the objective of improving the hover and manoeuvre performance. As indicated in chapter 1 this can probably be done without severe consequences for the drag at the high speed design condition. Such a modification process is one of trial and error in which the effects of a certain modification are analyzed by means of flow computations. Both inviscid flow and boundary layer calculations* are needed in this phase.

When the airfoil shape is completely fixed, the final step is to estimate the c_ℓ, c_d, c_m curves as a function of angle of attack and Mach number. This requires potential flow and boundary layer calculations, matched in an iterative cycle, as described in Section 6.1.1.

The three major steps of the design procedure are discussed in more detail in the following chapters.

*A Reynolds number of 5×10^6 was assumed for all viscous flow calculations and estimations throughout this report unless otherwise noted.

4 COMPUTATION OF A BASIC, SHOCK-FREE SHAPE BY MEANS OF HODOGRAPH THEORY

Nieuwland's hodograph theory (Ref. 6) is a method for the transformation of the incompressible potential flow around a lifting ellipse into a compressible potential flow around a lifting quasi-elliptical airfoil. In this theory four basic parameters appear, namely the free stream Mach number (Ma), the circulation of the flow Γ, the incidence α of the ellipse in the incompressible flow and a term ε_o in the expression $(1-\varepsilon_o)/(1+\varepsilon_o)$ for the thickness ratio of the ellipse.

These four parameter profiles are not closed at the rear end. From an engineering point of view the "gaps" are often negligibly small. If required, they can be closed by adding three more parameters, provided, that the physical interpretation is not destroyed by the appearance of limit lines in the supersonic region or a branch point outside the airfoil near the nose. Limit lines can also occur with the basic, four parameter profiles. For more details concerning the effect of the parameters on the airfoil geometry see references 8,17.

In the present investigation analytical closure of the airfoils was not envisaged because it was expected that the rear parts would have to be modified anyway for optimisation towards the hover and manoeuvre requirements. The initial parameter study was therefore limited to the four basic parameters mentioned above.

Of the four parameters the free stream Mach number can be chosen readily ; experience has shown that if a drag-rise Mach number of 0.85 is required, the Mach number for the theoretical shock-free design condition can be about 0.025 lower. The other parameters were varied such, that, not permitting c_ℓ to exceed 0.2, and avoiding limit lines in the supersonic regions, the largest possible amount of nose droop was obtained.

The best result of this first set of calculations was an airfoil designated NLR 7216. The nose shape of this airfoil is shown in figure 6. The pressure distribution in the high speed design condition bears strong resemblance to that given in figure 7. A characteristic feature of the airfoil is formed by a peak in the upper surface curvature distribution at $x/c = 0.067$ ($\sqrt{x/c} = 0.26$

in figure 8). This peak is caused by a limit line cusp just inside the airfoil and is of major importance for the shock-free upper surface flow.

It may be noted, that a somewhat similar, but less pronounced curvature distribution is exhibited by the FX69-H-098 airfoil. This is believed to be one reason for the comparatively good high speed performance of the FX69-H-098.

As mentioned before, such curvature distributions are also favourable for obtaining a high $c_{\ell_{max}}$ in the medium Mach number range. The reason is the following. At a certain high angle of attack and a certain Mach number, crest and curvature peak can coincide at the end of the supersonic zone. The expansion effects generated by the curvature peak then tend to decrease compressive effects and shock strengths at the crest. From another point of view the curvature peak promotes a hollow "peaky" suction loop at high angles of attack which, as shown by Pearcy (Ref.5), is favourable for reducing shock strength. It also reduces the crest pressure in such a way that the favourable effects occur at the desired Mach number. At the optimum condition the width of the supersonic suction region is, in fact, determined by the distance from the leading edge to the curvature peak ; obviously, the larger this distance, the higher $c_{\ell_{max}}$ can be. Apart from the "peaky" concept and the crest pressure criterium this is believed to be an important rule for obtaining a high $c_{\ell_{max}}$ in the medium Mach number range.*)

The rather forward position of the curvature peak of airfoil 7216 and, associated with this, the limited amount of nose droop, is believed to be the main reason for the disappointing manoeuvre and hover performance of this airfoil. (By means of crest pressure correlation $c_{\ell_{max}}$ at Ma=0.5 was estimated to be 1.05 and to much supercritical flow was indicated at the hover condition).

In order to remedy the situation the theory for quasi-elliptical airfoils was re-analysed and two new parameters λ_1 and λ_2 were

*) It should be mentioned, that curvature peaks of the type just mentioned could have an adverse effect on the wave drag in the hover condition (Ma=0.6, c_ℓ=0.65). It is therefore important to avoid, if possible, the appearance of supercritical flow regions at this condition.

introduced. The first parameter controls nose bluntness, the second controls the droop of the airfoils. With Ma, ε and T fixed at the values for airfoil 7216 the parameters λ_1 and λ_2 were systematically varied so as to give as much droop as possible while positioning the curvature peak (Fig.8) as rearward as possible. The parameter choice was in this case restricted by the appearance of limit lines.

The best result computed was airfoil 7223. For the geometrical improvement obtained compared to airfoil 7216 see figures 6 and 8. The shape of the sonic lines in the theoretical design condition (Ma=0.826, $c_\ell \approx 0.15$) are sketched in figure 7.

It is expected, that further improvement can still be obtained by systematically varying both the two new parameters and the four basic parameters. This could not be verified in the present investigation. Airfoil 7223 was therefore taken as the basic shock-free shape.

5 MODIFICATIONS TO THE BASIC SHAPE

5.1 General remarks

In modifying a basic, shock-free shape the important question is how large a modification can be tolerated if one does not want to lose the low-drag properties at the high speed design condition. Present NLR experience in this respect is, that in the first place, the modifications must be limited to the regions that have subsonic flow at the high speed design condition. Secondly, the location of the forward sonic points at the high speed design condition and the acceleration at these points must not be affected by the modification. The conditions at a rearward sonic point have appeared to be less critical (Ref.10).

For the basic airfoil of figure 7 the considerations just given suggest that the upper surface may be modified aft of approximately 70 % chord and the lower surface aft of 20 % chord. At the nose, experience is that modification is limited to the immediate vicinity of the position of the stagnation point at the high speed design condition.

So far, NLR have used the approximate potential flow method of

reference 18 to estimate the effect of contour modifications. In spite of the fact that this method is limited, in principle, to subcritical flow it has proved usefull in cases of supercritical, shock-free flow (Ref.10). In contrast to the hodograph method, the approximate method of Reference 18 predicts the subsonic flow field for a given shape. The method uses the surface singularity technique in combination with semi-empirical compressibility characteristics

For the purposes in mind, a dissipative type of finite difference method for the computation of transonic flows with shock waves would obviously be more appropriate. At the time of the present study such a method was not yet available at NLR. However a few check calculations for the high speed and manoeuvre design points by means of the method of Garabedian and Korn (Ref.19) could be made at the end of the investigation.*)

5.2 With pitching moment requirement (airfoil 1)

As indicated in chapters 2 and 3, modification of the basic high speed shape with the objective to optimize for hover and manoeuvre, must, in terms of pressure distributions, be directed towards the following.

i) In order to avoid shock waves on the upper surface at the hover condition and to have a high $c_{\ell_{max}}$ at Ma=0.5, the c_ℓ-value (c_ℓ^*) for which the flow first becomes critical in the 0.5 to 0.6 Mach number range must be as high as possible.

ii) To minimise boundary layer drag at the hover condition a laminar, accelerating flow of long extent is required on the lower surface at Ma=0.6, c_ℓ=0.65.

iii) To reduce the shock strength at high c_ℓ at Ma=0.5, improve, if possible, the "peakiness" of the suction loop.

iv) To satisfy the pitching moment requirement reduce, if necessary, the load near the trailing edge.

v) To reduce the risk of early boundary layer separation at the manoeuvre and high speed conditions avoid large pressure gradients and apply, if possible, a Stratford (Ref.12) type of pressure recovery.

Of these directives iii) stands rather isolated in the sense that a limited modification of the nose shape, with the objective

*) Most of these were performed by BHC

of improving the "peakiness" of the suction loop, does not interfere with the other directives. Considering the incompressible suction loop at the optimum angle of attack in figure 9 suggests to strive after a somewhat more "hollow" suction loop. With the limitation to restrict nose shape modification to the immediate vicinity of the stagnation point location at the high speed design condition, this could only be realised by shifting the suction peak (Fig. 9). The corresponding leading edge modification has been indicated in figure 6.

i) implies that the freedom to modify the rear parts of the basic airfoil should be used to increase the loading in that region. This, however, is in conflict with the pitching moment requirement and also with ii) and v), which means that a compromise must be found. This was realised by a trial and error process in which the effect of several modifications on the pressure distribution at the various design points was calculated by means of the method of reference 18. The methods of Thwaites (Ref. 20) and Nash (Ref. 21) were used to check the boundary layer behaviour at the most critical conditions. Transition was predicted by means of the Michel/Smith criterium (Ref. 22).

As a final shape the one shown in figures 10 and 11 was considered acceptable. Note that the upper surface was modified aft of 70 % x/c and the lower surface aft of 50 %. Tabulated coordinates of the airfoil are given in table 2. The predicted aerodynamic characteristics are discussed in section 6.1.

5.3 Without pitching moment requirement (airfoil 2)

Apart from the pitching moment requirement iv), the directives listed in the beginning of this section apply equally well in the case without pitching moment requirement.

The optimisation process for airfoil 2, leads nevertheless to a characteristic difference between the two airfoils. This is caused by the fact that the absence of a pitching moment requirement allows application of the rear-loading concept.

In the case of airfoil 1 some negative loading was needed near the trailing edge to keep the pitching moment within the required limits. As a consequence some wave drag had to be tolerated in the hover condition (Fig. 14). A further consequence of this is that

a fully laminar lower surface boundary layer is required to obtain an acceptable L/D value.

In contrast, the application of (positive) rear loading increases the critical c_l-value in the 0.5 - 0.6 Mach number range. As a result the wave drag at hover is negligible (Fig. 33). However, the boundary layer drag is somewhat higher than that of airfoil 1 due to a more forward transition point on the lower surface. As a result the L/D values of the two airfoils do not differ very much. Thus, the benefits of the rear-loading concept are mainly limited to a higher $c_{l_{max}}$. Figures 29 and 30 present the airfoil shape that has resulted from the trial and error process for airfoil 2. Tabulated coordinates are given in table 3. Section 6.2 discusses the predicted aerodynamic characteristics.

6 PREDICTED AERODYNAMIC CHARACTERISTICS

6.1 Airfoil 1

6.1.1 The high speed, hover and manoeuver design points

Aerodynamic data relevant to the high speed, hover and manoeuver design points are presented in figures 12 to 18.

Figure 12 presents a comparison of the hodograph pressure distribution for the basic airfoil at the design Mach number of 0.826, with that for the modified airfoil calculated by means of the approximate subsonic method of reference 18. On the basis of this comparison it is expected that the modification does not give rise to significant wave drag at the design Mach number of 0.826.

Figure 13 presents a result of the Carabedian/Korn method for Ma = 0.85, c_l 0. The predicted wave drag of 0.005 suggests that the drag coefficient at this condition will be just below the required value of 0.013 provided that the boundary layer can negotiate the pressure rise at the rear also in the presence of a shock wave.

The pressure distribution at the hover condition, including the effect of the boundary layer, is given in figure 14. It was calculated by means of the method of reference 23. This method uses a single computer program and combines the analyses of references 18, 20, 21, and the transition criterium of reference 22 in an iterative cycle. The Square and Young formula (Ref. 26) is used

to calculate the drag from the boundary layer properties at the trailing edge. The basis of the method is described in reference 24. As shown in figure 14 the flow is supercritical. Based on correlation with the FX69-H-098 airfoil the wave drag is estimated to be 0.001. The total drag is estimated to be 0.0074. This leads to a lift/drag ratio of 90 which is close to that of the FX69-H-098 airfoil.

With the present state of the art in theoretical aerodynamics accurate prediction of $c_{\ell_{max}}$ is not yet possible, so that one is limited to empirical estimates. Since $c_{\ell_{max}}$ at Ma = 0.5 is limited by shock-induced stall, two methods are available to aid in estimating this parameter. Both methods, however, are based on the assumption that the stall mechanism for the new sections are similar to that of the FX69-H-098 airfoil. With the first method incremental values of $c_{\ell_{max}}$ may be estimated by observing the minimum pressure as a function of c_ℓ at the critical pressure value (Fig. 15). The second method utilizes Sinnott's criterium (Ref. 25) in relation to the crest pressure expressed as a function of c_ℓ (Fig. 16). Both methods suggest that $c_{\ell_{max}}$ at Ma = 0.5 of airfoil 1 will be approximately 0.1 below that of the Wortmann FX69-H-098 airfoil. This would bring it in the 1.20 to 1.25 range for a Reynolds number of 4 to 5 million.

A similar conclusion is obtained from pressure distributions calculated by means of the Garabedian/Korn method (Fig. 17). The test results for the FX69-H-098 airfoil suggest that at Re = 4×10^6, for the specific type of pressure distribution considered, shock induced separation limits a further increase of c_ℓ when the local pressure coefficient just in front of the shock exceeds the value -4 (Fig. 4). According to figure 17, this would, in inviscid flow, be obtained for $c_\ell \approx 1.28$. With a 5% viscous lift loss (suggested by the difference between potential flow and experiment in figures 15, 16) this leads also to a $c_{\ell_{max}}$ of 1.20 to 1.25.

An assumption underlying the considerations just given, has been that there is no drastic difference in the rear separation characteristics between the FX69-H-098 and the present airfoil. The results of boundary layer calculations for subcritical conditions, presented in figure 18 in terms of the local skin-friction coefficient at x/c = 0.95, suggest that this is indeed the case.

6.1.2 Estimated c_ℓ, c_d and c_m-curves

In the subcritical regime lift and drag value (Fig. 19, 20, and 21) were calculated by means of the method of reference 23. The method uses the Square and Young formula (Ref. 26) to calculate the drag from the boundary layer properties at the trailing edge. Data for supercritical flow were obtained by extrapolation of subcritical values, using the test results of the FX69-H-098 and some isolated Garabedian/Korn data points as a basis.

The estimated lift and drag boundaries in the c_ℓ-Ma plane are summarized in figure 22. The drag boundary was obtained from figures 20 and 21 whereas the maximum lift boundary was estimated by means of minimum pressure and crest pressure correlation as described above.

Finally, calculated pitching moment curves are presented in figures 23, 24 and pressure distributions for several subcritical flow conditions in figures 25 to 28.

6.2 Airfoil 2

A similar set of data for airfoil 2 is given in figures 31 to 47. These do not need further discussion because they were obtained in the same way as those for airfoil 1.

As indicated by figure 32 a slightly higher wave drag must be expected at the high speed (Ma=0.85), low c_ℓ condition. This would bring the Mach number for which c_d = 0.013 just below Ma = 0.85.

Figure 33 presents the pressure distribution at the hover condition. The lift/drag ratio is estimated to be 94.

Information relevant to $c_{\ell\,max}$ at Ma=0.5 is given by figures 34 to 38. These suggest that $c_{\ell\,max}$ will be between 1.25 and 1.30 at Ma=0.5.

Further data concerning lift, drag, pitching moment and pressure distributions is given by figures 38 to 47.

7 CONCLUDING REMARKS

A design study for two rotor airfoils has been performed using as a basis the hodograph method for the computation of transonic, shock-free flow. The results of this study are summarized in table 4 in terms of predicted aerodynamic data of two new airfoils for the manoeuvre, hover and high speed design points.

As compared with a contemporary rotor airfoil, designed along more empirical lines, the high speed performance predicted for the new airfoils is substantially better. This has been obtained at the cost of a slight loss in $c_{\ell_{max}}$ in the medium Mach number range (manoeuvre condition). However, $c_{\ell_{max}}$ at Ma=0.5 is expected to be 20 to 30 % higher still than that of the standard NACA 0012 airfoil.

Of the two airfoils one satisfies the usual requirement for a small pitching moment and one does not. It appears that the main implication of the pitching moment requirement is a reduction of $c_{\ell_{max}}$ in the medium Mach number range.

A characteristic feature of the airfoils is formed by the upper surface curvature distribution, which exhibits a peak at x/c 0.09. It is believed that the presence of such a curvature peak is an essential feature of airfoils that must combine high speed and $c_{\ell_{max}}$ performance in the way required for application in a helicopter rotor. At high speed, the curvature peak triggers a favourable "peaky" type of flow over most of the upper surface. In the manoeuvre the curvature peak is instrumental in decreasing the crest pressure to the level required for obtaining a high $c_{\ell_{max}}$ at Ma 0.05. At the same time it determines the shock position and through this the chordwise extent of the region of supersonic flow. A high manoeuvre $c_{\ell_{max}}$ therefore requires that the position of the curvature peak be as far from the nose as possible.

In the present investigation the possibilities with respect to curvature peak position offered by the hodograph method could be explored only partially. Further parameter studies are required to answer the question whether it could be shifted beyond x/c=0.09. In case of continuation of the present work this would be one subject for further research. Another, necessary step of a follow-on program would

obviously be the careful two-dimensional transonic testing of the two new airfoils described in this report. Preferably, this should be done under both static and dynamic conditions.

8 REFERENCES

1	Wortmann, F.X. and Drees, Jan M.	Design of airfoils for rotors Paper presented at CAL/AVLABS 1969 Symposium on Aerodynamics of Rotary Wing and VTOL Aircraft
2	Benson, R.G., Dadone, L.U., Gormont, R.E. and Kohler, G.R.	Influence of airfoils on stall flutter boundaries of articulated helicopter rotors Paper presented at the 28th annual forum of the American Helicopter Society, Washington D.C., May 1972
3	Pearcey, H.H., Wilby, P.G., Riley, M.J. and Brotherhood, P.	The derivation and verification of a new rotor profile on the basis of flow phenomena; airfoil research and flight tests Paper presented at the AGARD Specialists Meeting on "The Aerodynamics of Rotary Wings", Marseille, 13-15 Sept. 1972 (RAE Tech. Memo Aero 1440)
4	Reichert, G. and Wagner, S.N.	Some aspects of the design of rotor airfoil shapes Paper presented at the AGARD Specialists Meeting on "The Aerodynamics of Rotary Wings", Marseille, 13-15 Sept. 1972 AGARD-CPP-III, Paper 14
5	Pearcy, H.H.	The Aerodynamic Design of Section Shapes for Swept Wings Advances in Aeronautical Sciences, Vol. 3 Pergamon Press, London, 1962

6 Nieuwland, G.Y. — Transonic Potential Flow Around A Family of Quasi-elliptical Aerofoil Sections
NLR TR T.172, 1967

7 Spee, B.M. and Uylenhoet, R. — Experimental Verification of Shock-free Transonic Flow Around Quasi-elliptical Aerofoil Sections
Lecture presented at the 6th ICAS Congress, Munich, September 1968
(Also NLR MP 68003 U)

8 Boerstoel, J.W. and Uylenhoet, R. — Lifting Aerofoils with Supercritical Shockless Flow
Lecture presented at the 7th ICAS Congress Roma, September 1970, Paper 70-15
(Also NLR MP 70015 U)

9 Spee, B.M. — Investigations on the transonic flow around aerofoils
NLR TR 69122 U, 1969

10 Loeve, W. and Slooff, J.W. — On the use of "panel methods" for predicting subsonic flow about aerofoils and aircraft configurations
Paper presented at the 4.Jahrestagung of DGLR
Baden-Baden, Germany, October 1971
(Also NLR MP 71018 U)

11 — Proposal for an analytical study for the design of advanced rotor airfoils
Bell Helicopter Company
Document 299-199-046, March 1971

12 Stratford, B.S. An Fxperimental Flow with Zero Skin Friction throughout Its Region of Pressure Rise
J.F.M. Volume 5, January 1959, pp. 17-35

13 Smith, A.M.O. Aerodynamics of High-Lift Airfoil Systems
Paper presented at the AGARD Specialists' Meeting on the "Fluid Dynamics of Aircraft Stalling"
Lisbon, 25-28 April, 1972
AGARD-CPP-102, Paper 10

14 Boerstoel, J.W. A Survey of Symmetrical Transonic Potential Flows Around Quasi-elliptical Aerofoil Sections
NLR TR T.136, 1967

15 Unplublished Data
Bell Helicopter Company

16 Wortmann, F.X. Design of Airfoils with High Lift at Low and Medium Subsonic Mach Numbers
Paper presented at the AGARD Specialists' Meeting on the "Fluid Dynamics of Aircraft Stalling"
Lisbon, 25-28 April, 1972
AGARD-CPP-102, Paper 7

17 Boerstoel, J.W, and Huizing, G.H. ALGOL-programs for the computation of quasi-elliptical, shock-free, transonic airfoils
NLR TR 72128C, 1972

18 Labrujere, Th.E., Loeve, W. and Slooff, J.W. An approximate method for the calculation of the pressure distribution on wing-body combinationsat subcritical speeds
AGARD C.P. No.71, paper 11, 1970

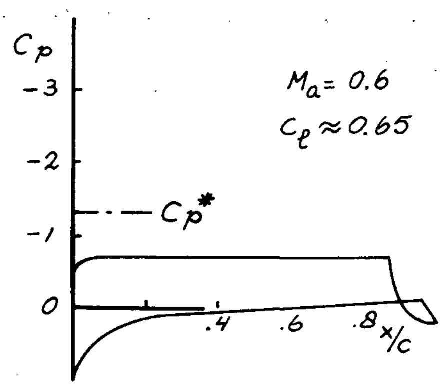
19	Bauer, F., Garabedian, P. and Korn, D.	Supercritical Wing Sections Springer, Berlin, 1972
20	Thwaites, B.	Approximate calculation of the laminar boundary layer Aero Quart., Vol. 1, 1949
21	Nash, J.F. and McDonald, A.G.J.	Calculation of momentum thickness in a turbulent boundary layer at Mach number upto unity NPL Aero Rept. 1207, 1966
22	Cebeci, T.J., Mosinskis, G.J. and Smith, A.M.O.	Calculation of viscous drag of two-dimensional and axis-symmetric bodies in incompressible flow AIAA Paper 72-1, 1972
23	Piers, W.J. and Slooff, J.W.	Unpublished work NLR
24	Piers, W.J. and Slooff, J.W..	Calculation of the displacement effect in two-dimensional, subsonic, attached flow ; examples of calculations using measured displacement thicknesses NLR TR 72116, to be published
25	Sinnott, C.S.	Estimation of the transonic characteristics of aerofoils NPL Aero Rept. 385, 1959
26	Squire, H.B. and Young, A.D.	The calculation of the profile drag of aerofoils A.R.C. R+M 1838, 1937

APPENDIX

SOME REMARKS ON POSSIBLE UPPER LIMITS FOR ROTOR AIRFOIL PERFORMANCE AT THE SEPARATE DESIGN POINTS

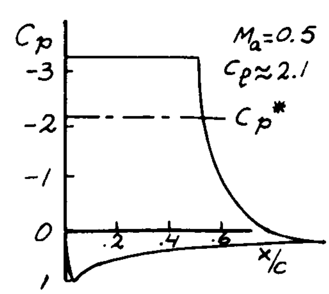
For comparative purposes it may be illustrative to discuss shortly (by lack of hard facts in a somewhat speculative way) the upper limits of hover, manoeuvre and high speed capability of airfoils optimised for each of these flight conditions separately.

Optimisation for the hover condition only would probably lead to a laminar subsonic "rooftop" upper surface pressure distribution, followed by a Straford (Ref.12) type of pressure recovery to the trailing edge (Sketch 1). Reference 13 suggests that the extent of the rooftop might go as far as 90 % chord for Re = 5×10^6 before trailing edge separation occurs. With some kind of laminar (accelerating) flow on the lower surface it is estimated that c_d-values as low as 0.0030 could be obtained. This would lead to L/D values as high as 220. However, the manoeuvre and high speed qualities would be poor as trailing edge separation would occur almost instantaneously when the angle of attack or Mach number would be increased beyond the typical values for hover.



Sketch 1 Optimal pressure distribution for high L/D at hover

The work of Liebeck, as reported by A.M.O. Smith in reference 13, provides a basis for an estimate of the upper limit of $c_{\ell_{max}}$ at Ma=0.5. As in the hover case a laminar, rooftop type of pressure distribution followed by a Stratford pressure recovery appears to be the best. In compressible flow the rooftop suction level would seem to be limited by the onset of shock-induced separation. Assuming that this occurs when the local Mach number exceeds 1.25 (which is supported by experimental evidence), $c_{\ell_{max}}$



Sketch 2 Optimal pressure distribution for $c_{\ell_{max}}$ at Ma=.5

values upto 2.1 would seem possible at $Ma=0.5$ and $Re=5x10^6$. This would be obtained for a rooftop extent of 50-60 % of the chord (Sketch 2).

The fact, that "rooftop" type of pressure distributions, although differing in extent and suction level, appear suitable for both hover and manoeuvre, suggests a certain amount of compatibility between these two conditions. However, looking at the rather unusual shapes that are obtained (Ref.13), there is little hope for acceptable high speed characteristics.

The highest drag-rise Mach number at zero lift is, of course, obtained with the flat plate at zero angle of attack. With constructional constraints coming-in, the lower limit for t/c is probably something like 4 %. Experience with quasi-elliptical airfoils (Ref.14) then leads to a drag rise Mach number of approximately 0.93 as an upper limit.

Flight Condition	Specific	Quantity	NACA 0012 *)	Wortmann FX69-H-098 *)	Tentative requirements in present design study	Probable upper limits for separate optimisation +)
hover	$Ma = 0.6$	c_ℓ/c_d	85	≈ 95	100	≈ 220
	$c_\ell = 0.65$	c_m	≈ 0	-0.016	$<\lvert 0.02\rvert$ / no reqmt	?
Manoeuvre	$Ma \approx 0.5$	$c_{\ell_{max}}$	1.01	1.33	> 1.35	≈ 2.1
High Speed	$c_\ell \approx 0$, $c_d = 0.013$	Ma	0.77	0.80	> 0.85	≈ 0.93
Other conditions		Re	3to6x10^6	3to6x10^6	5x10^6	5x10^6
		$(t/c)_{max}$ %	12	9.8	> 4 , < 15	> 4

*) Measurements UAC Tunnel

+) Estimated, see Appendix

Table 1 Comparison of tentative requirements for design points with characteristics of baseline airfoils

AIRFOIL 1					
x	z	x	z	x	z
+1.00000	-0.00000	+0.17401	+0.05074	+0.06154	-0.01754
+0.98674	+0.00161	+0.16260	+0.05004	+0.06978	-0.01840
+0.97405	+0.00345	+0.14943	+0.04915	+0.07732	-0.01913
+0.94056	+0.00820	+0.13619	+0.04815	+0.08778	-0.02003
+0.90954	+0.01262	+0.12628	+0.04730	+0.09531	-0.02072
+0.87705	+0.01772	+0.11871	+0.04662	+0.10662	-0.02162
+0.85176	+0.02196	+0.11493	+0.04624	+0.11645	-0.02234
+0.82615	+0.02614	+0.10901	+0.04563	+0.12776	-0.02313
+0.79864	+0.03053	+0.10522	+0.04521	+0.13908	-0.02387
+0.76970	+0.03505	+0.10145	+0.04476	+0.15039	-0.02455
+0.73974	+0.03909	+0.09955	+0.04453	+0.16170	-0.02520
+0.71085	+0.04192	+0.09766	+0.04428	+0.17490	-0.02589
+0.67758	+0.04438	+0.09577	+0.04401	+0.18864	-0.02656
+0.66250	+0.04539	+0.09457	+0.04384	+0.20185	-0.02714
+0.64741	+0.04635	+0.09337	+0.04365	+0.21505	-0.02768
+0.62960	+0.04740	+0.09217	+0.04346	+0.22825	-0.02817
+0.61263	+0.04833	+0.08977	+0.04307	+0.24145	-0.02862
+0.59600	+0.04919	+0.08617	+0.04243	+0.25655	-0.02908
+0.58091	+0.04991	+0.07672	+0.04057	+0.27168	-0.02949
+0.56635	+0.05055	+0.06728	+0.03842	+0.28678	-0.02985
+0.54748	+0.05133	+0.05969	+0.03648	+0.31696	-0.03043
+0.53239	+0.05189	+0.05219	+0.03435	+0.34904	-0.03083
+0.51353	+0.05254	+0.04498	+0.03207	+0.38098	-0.03107
+0.49485	+0.05310	+0.03956	+0.03017	+0.41307	-0.03115
+0.47976	+0.05351	+0.02729	+0.02515	+0.44726	-0.03106
+0.46137	+0.05394	+0.01748	+0.02005	+0.48125	-0.03078
+0.44628	+0.05425	+0.00978	+0.01508	+0.51389	-0.03030
+0.42929	+0.05455	+0.00416	+0.01023	+0.54720	-0.02959
+0.41232	+0.05478	+0.00125	+0.00617	+0.58067	-0.02861
+0.39533	+0.05497	+0.00000	+0.00000	+0.61492	-0.02754
+0.38024	+0.05508	+0.00054	-0.00251	+0.64771	-0.02638
+0.36513	+0.05514	+0.00118	-0.00363	+0.67813	-0.02512
+0.34814	+0.05516	+0.00265	-0.00520	+0.70940	-0.02375
+0.33421	+0.05512	+0.00450	-0.00641	+0.74105	-0.02238
+0.31722	+0.05501	+0.00730	-0.00765	+0.76943	-0.02097
+0.30211	+0.05486	+0.00992	-0.00860	+0.79937	-0.01947
+0.28701	+0.05465	+0.01334	-0.00964	+0.82514	-0.01805
+0.27184	+0.05438	+0.01749	-0.01072	+0.85298	-0.01653
+0.25673	+0.05404	+0.02181	-0.01170	+0.87654	-0.01505
+0.24162	+0.05363	+0.02769	-0.01286	+0.90896	-0.01250
+0.22738	+0.05317	+0.03500	-0.01411	+0.94070	-0.00885
+0.21605	+0.05275	+0.03938	-0.01477	+0.97353	-0.00352
+0.20093	+0.05212	+0.04638	-0.01573	+0.98677	-0.00135
+0.18724	+0.05146	+0.05274	-0.01654	+1.00000	-0.00000

Table 2 Coordinates of airfoil 1.

AIRFOIL 2					
x	z	x	z	x	z
+1.00000	-0.00000	+0.14953	+0.04884	+0.09527	-0.02091
+0.98674	+0.00241	+0.13629	+0.04787	+0.10658	-0.02184
+0.97405	+0.00479	+0.12638	+0.04704	+0.11641	-0.02258
+0.94057	+0.01037	+0.11881	+0.04637	+0.12772	-0.02339
+0.90956	+0.01486	+0.11503	+0.04601	+0.13903	-0.02415
+0.87707	+0.01953	+0.10911	+0.04540	+0.15034	-0.02486
+0.85180	+0.02313	+0.10532	+0.04499	+0.16165	-0.02553
+0.82620	+0.02672	+0.10154	+0.04456	+0.17485	-0.02625
+0.79870	+0.03025	+0.09965	+0.04432	+0.18859	-0.02695
+0.76978	+0.03379	+0.09775	+0.04408	+0.20179	-0.02755
+0.73983	+0.03724	+0.09586	+0.04382	+0.21499	-0.02812
+0.71095	+0.04031	+0.09466	+0.04364	+0.22819	-0.02864
+0.67768	+0.04299	+0.09346	+0.04346	+0.24140	-0.02911
+0.66260	+0.04403	+0.09226	+0.04327	+0.25649	-0.02961
+0.64752	+0.04502	+0.08986	+0.04288	+0.27162	-0.03005
+0.62970	+0.04611	+0.08626	+0.04225	+0.28672	-0.03044
+0.61274	+0.04707	+0.07681	+0.04041	+0.30369	-0.03082
+0.59610	+0.04796	+0.06736	+0.03828	+0.31690	-0.03108
+0.58101	+0.04871	+0.05977	+0.03636	+0.33200	-0.03133
+0.56646	+0.04939	+0.05226	+0.03425	+0.34898	-0.03155
+0.54760	+0.05020	+0.04505	+0.03198	+0.36582	-0.03171
+0.53251	+0.05080	+0.03962	+0.03009	+0.38092	-0.03182
+0.51365	+0.05148	+0.02735	+0.02509	+0.39602	-0.03187
+0.49496	+0.05208	+0.01753	+0.02002	+0.41301	-0.03187
+0.47987	+0.05253	+0.00981	+0.01506	+0.44720	-0.03169
+0.46149	+0.05299	+0.00418	+0.01022	+0.48119	-0.03125
+0.44640	+0.05333	+0.00127	+0.00617	+0.51383	-0.03061
+0.42941	+0.05367	+0.00000	+0.00000	+0.54714	-0.02968
+0.41244	+0.05394	+0.00054	-0.00251	+0.58061	-0.02827
+0.39545	+0.05416	+0.00117	-0.00363	+0.61485	-0.02580
+0.38036	+0.05430	+0.00264	-0.00521	+0.64763	-0.02178
+0.36525	+0.05439	+0.00449	-0.00642	+0.67804	-0.01761
+0.34826	+0.05444	+0.00728	-0.00767	+0.70929	-0.01370
+0.33433	+0.05443	+0.00991	-0.00862	+0.74093	-0.01017
+0.31734	+0.05436	+0.01332	-0.00967	+0.76930	-0.00755
+0.30223	+0.05424	+0.01747	-0.01076	+0.79924	-0.00535
+0.28713	+0.05406	+0.02178	-0.01174	+0.82501	-0.00391
+0.27195	+0.05382	+0.02766	-0.01292	+0.85286	-0.00262
+0.25685	+0.05351	+0.03497	-0.01418	+0.87643	-0.00191
+0.24173	+0.05313	+0.03935	-0.01485	+0.90886	-0.00102
+0.22749	+0.05271	+0.04635	-0.01583	+0.94063	-0.00043
+0.21616	+0.05231	+0.05270	-0.01664	+0.97351	-0.00005
+0.20104	+0.05171	+0.06151	-0.01767	+0.98676	+0.00005
+0.18734	+0.05107	+0.06975	-0.01854	+1.00000	-0.00000
+0.17412	+0.05038	+0.07728	-0.01929		
+0.16276	+0.04971	+0.08774	-0.02026		

Table 3 Coordinates of airfoil 2.

Flight Condition	Specific	Quantity	NACA 0012 *)	Worthmann FX69-H-098 *)	Airfoil 1 (NLR 7223-62) +)	Airfoil 2 (NLR 7223-43) +)
Hover	$Ma = 0.6$	c_ℓ/c_d	85	≈ 95	91	94
	$c_\ell = 0.65$	c_m	≈ 0	- 0.016	- 0.019	- 0.065
Manoeuvre	$Ma \approx 0.5$	$c_{\ell_{max}}$	1.01	1.33	≈ 1.25	≈ 1.30
High Speed	$c_\ell \approx 0$, $c_d = 0.013$	Ma	0.77	0.80	0.85	0.85
Other conditions	Re		$3 \text{to} 6 \times 10^6$	$3 \text{to} 6 \times 10^6$	5×10^6	5×10^6
	$(t/c)_{max}$%		12	9.8	8.6	8.6

*) Measurements UAC Tunnel
+) Expected

Table 4 Comparison of expected characteristics at design points with baseline airfoil data

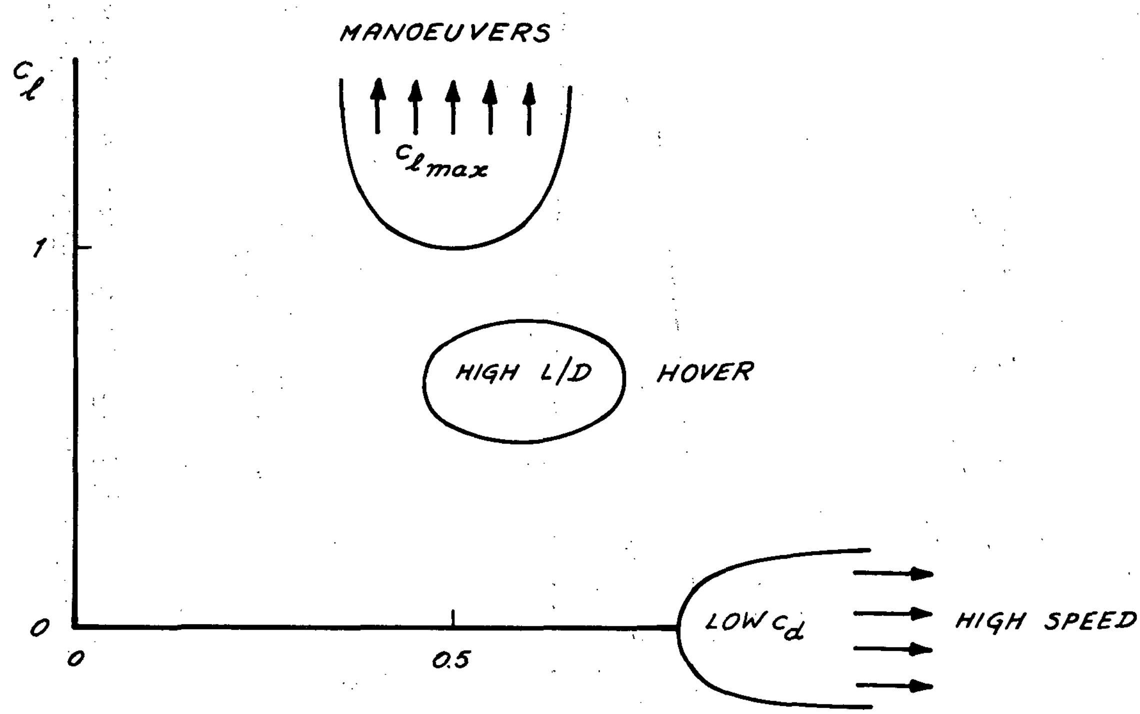


FIG. 1 THE CRITICAL REGIONS FOR A ROTOR AIRFOIL.

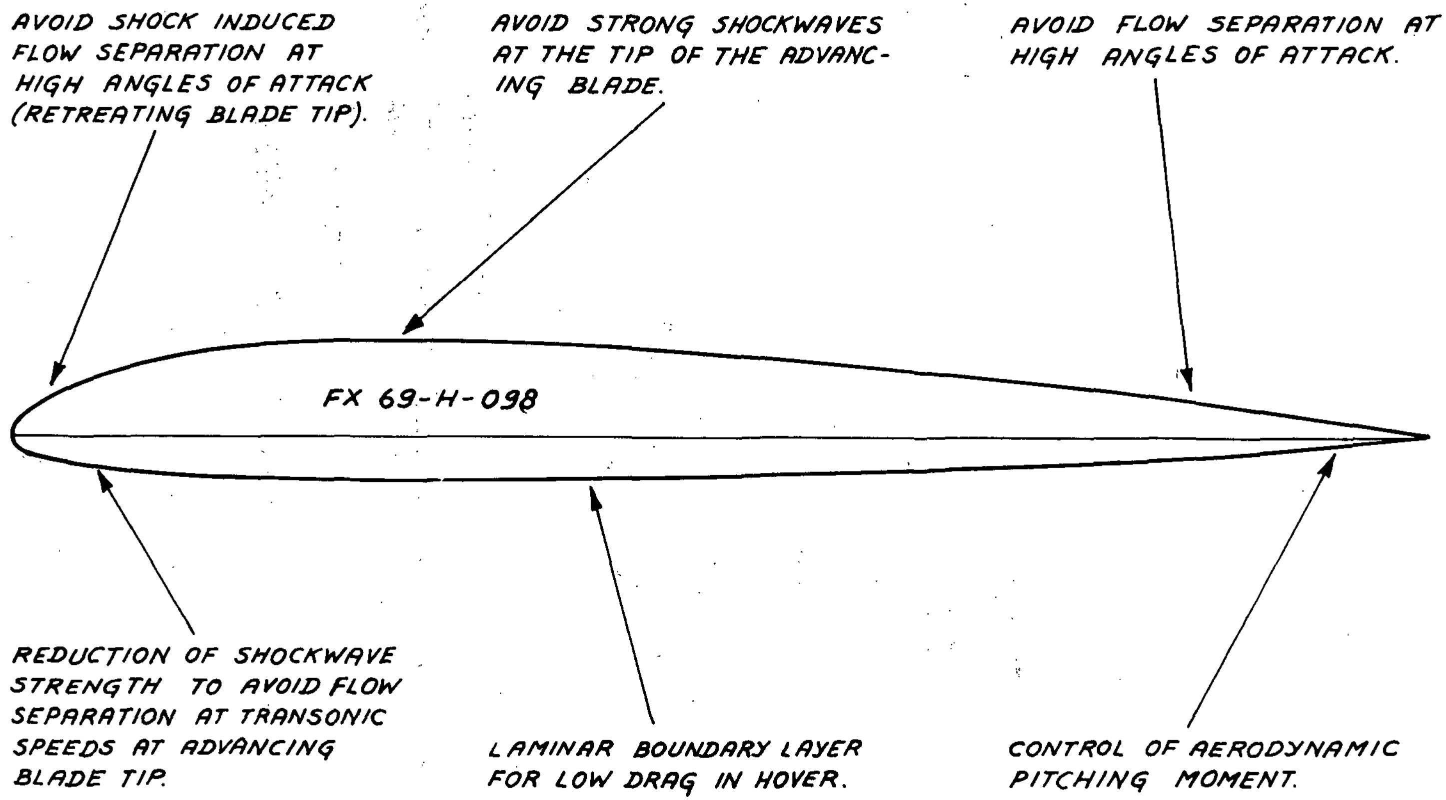
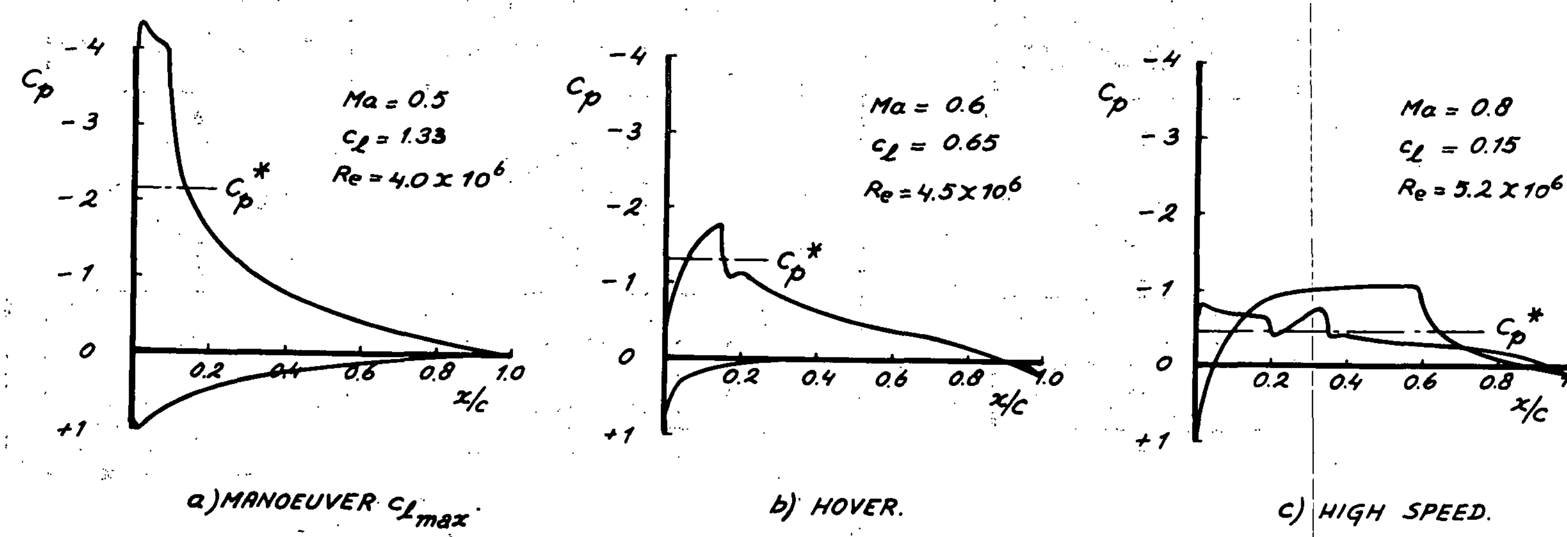


FIG. 2 DESIGN CONSIDERATIONS USED IN SHAPING THE BASELINE AIRFOIL (WORTMANN FX69-H-098).



a) MANOEUVER $c_{\ell_{max}}$

b) HOVER.

c) HIGH SPEED.

FIG. 3 PRESSURE DISTRIBUTIONS AT DESIGN POINTS FOR FX69-H-098 AIRFOIL.

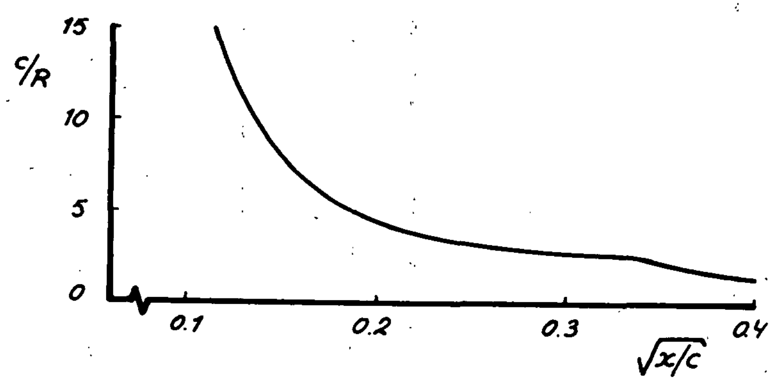


FIG. 4 UPPER SURFACE CURVATURE DISTRIBUTION FOR FX69-H-098 AIRFOIL.

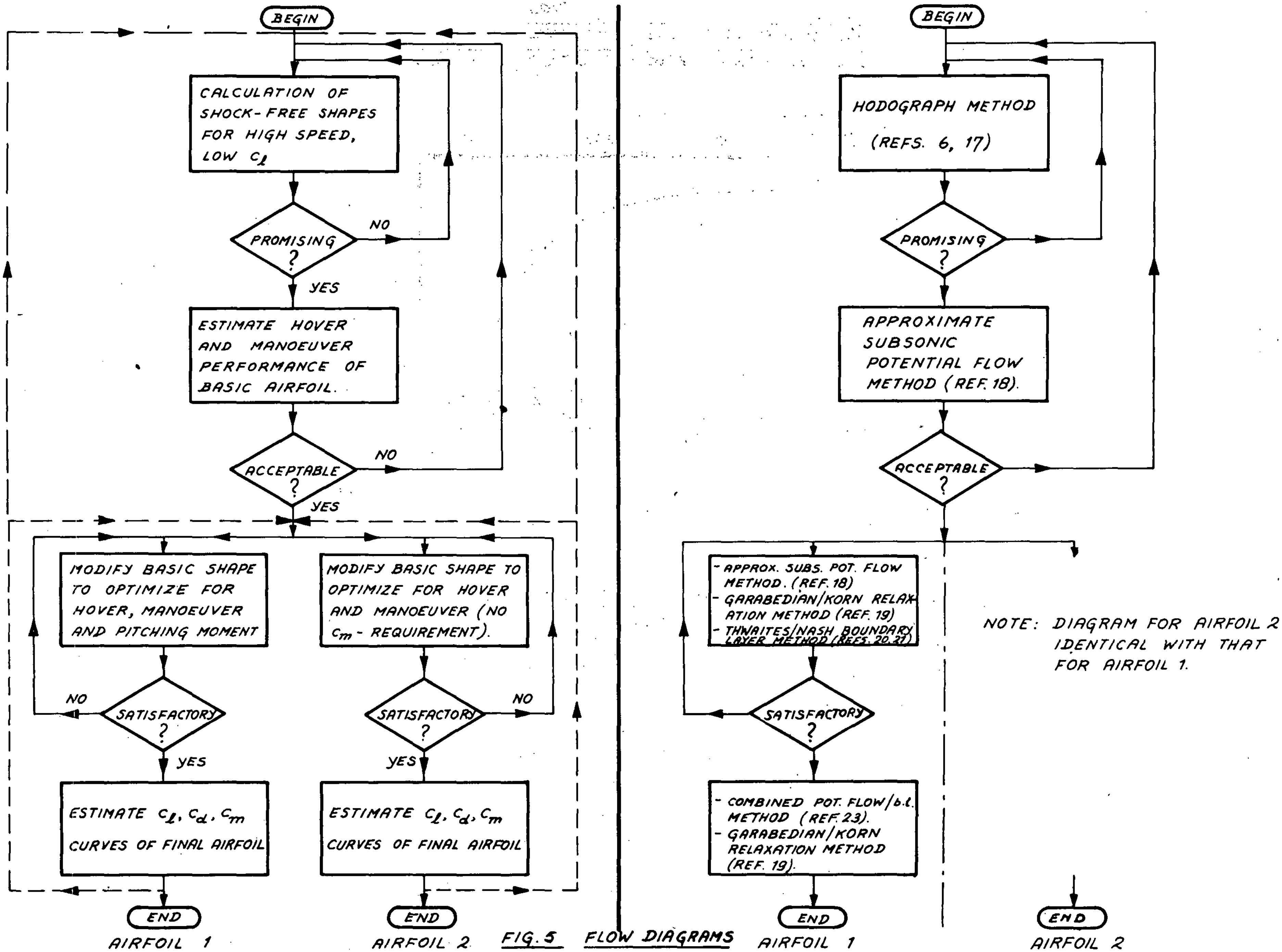


FIG. 5 FLOW DIAGRAMS OF DESIGN PROCEDURE.

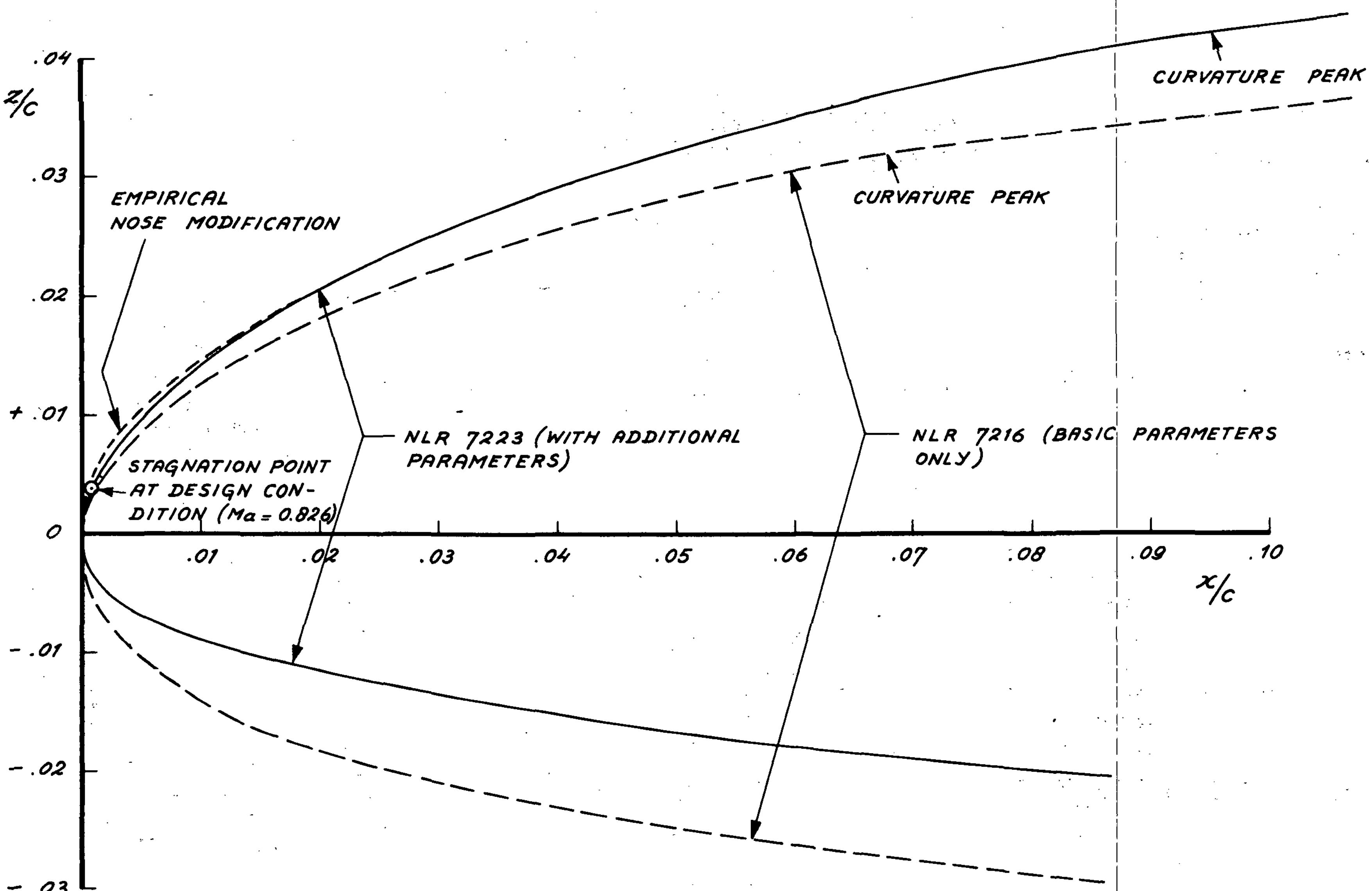


FIG. 6 COMPARISON OF NOSE SHAPES OBTAINED WITH AND WITHOUT ADDITIONAL DESIGN PARAMETERS.

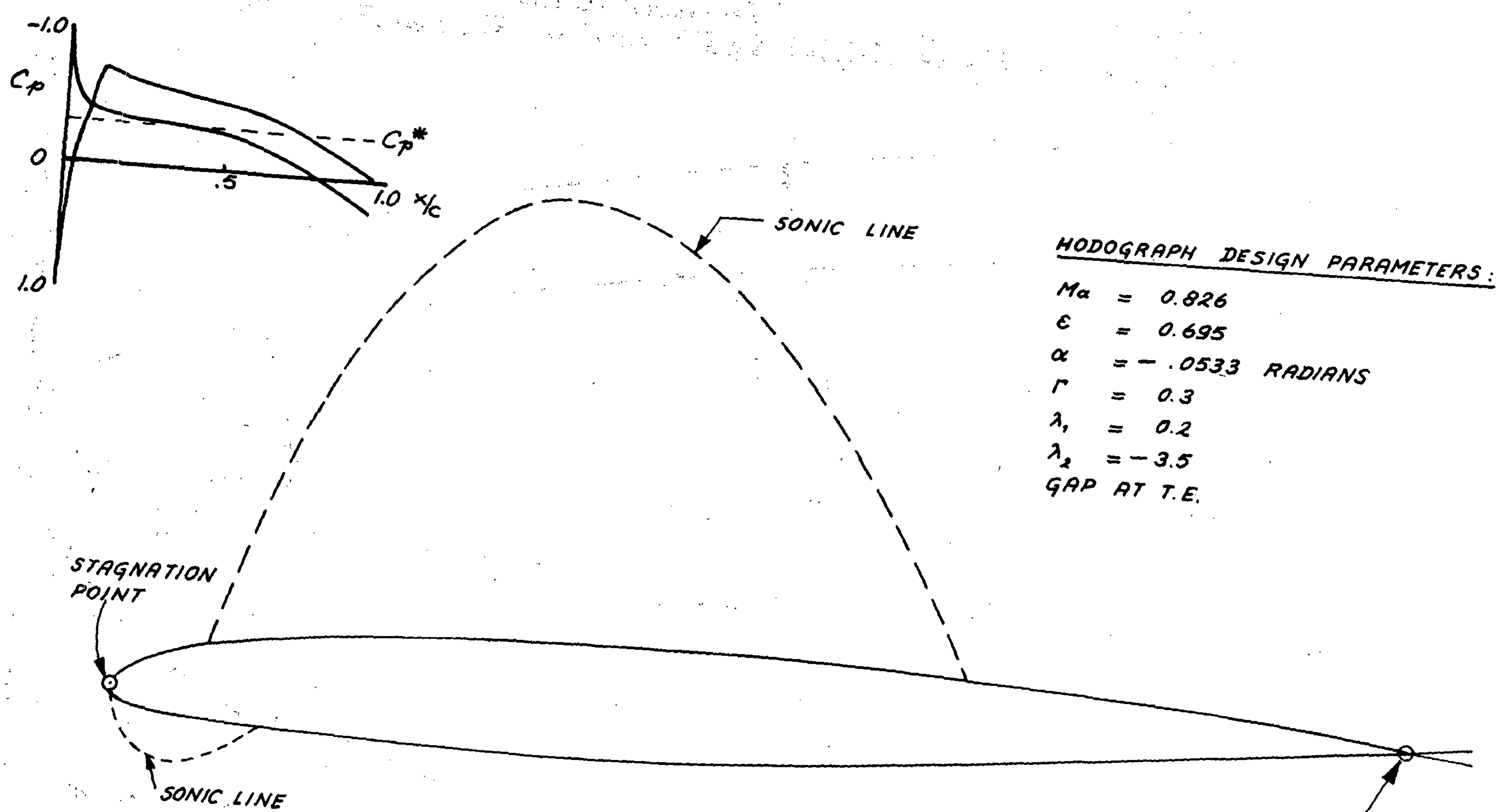


FIG. 7 NLR 7223 BASIC SHAPE, SONIC LINES AND PRESSURE DISTRIBUTION AT HIGH SPEED DESIGN CONDITION.

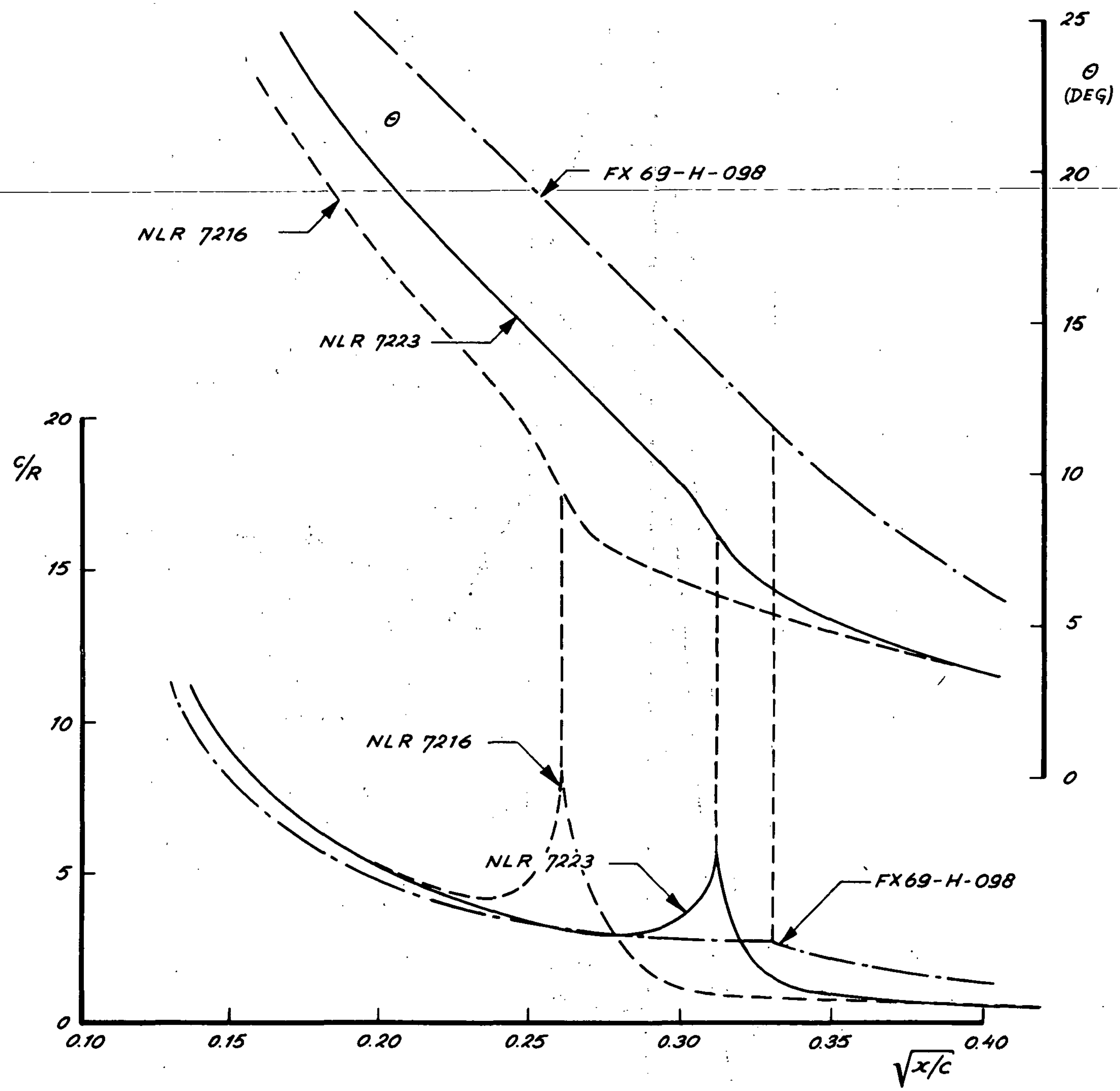


FIG. 8 COMPARISON OF UPPER SURFACE CURVATURE AND SLOPE DISTRIBUTIONS.

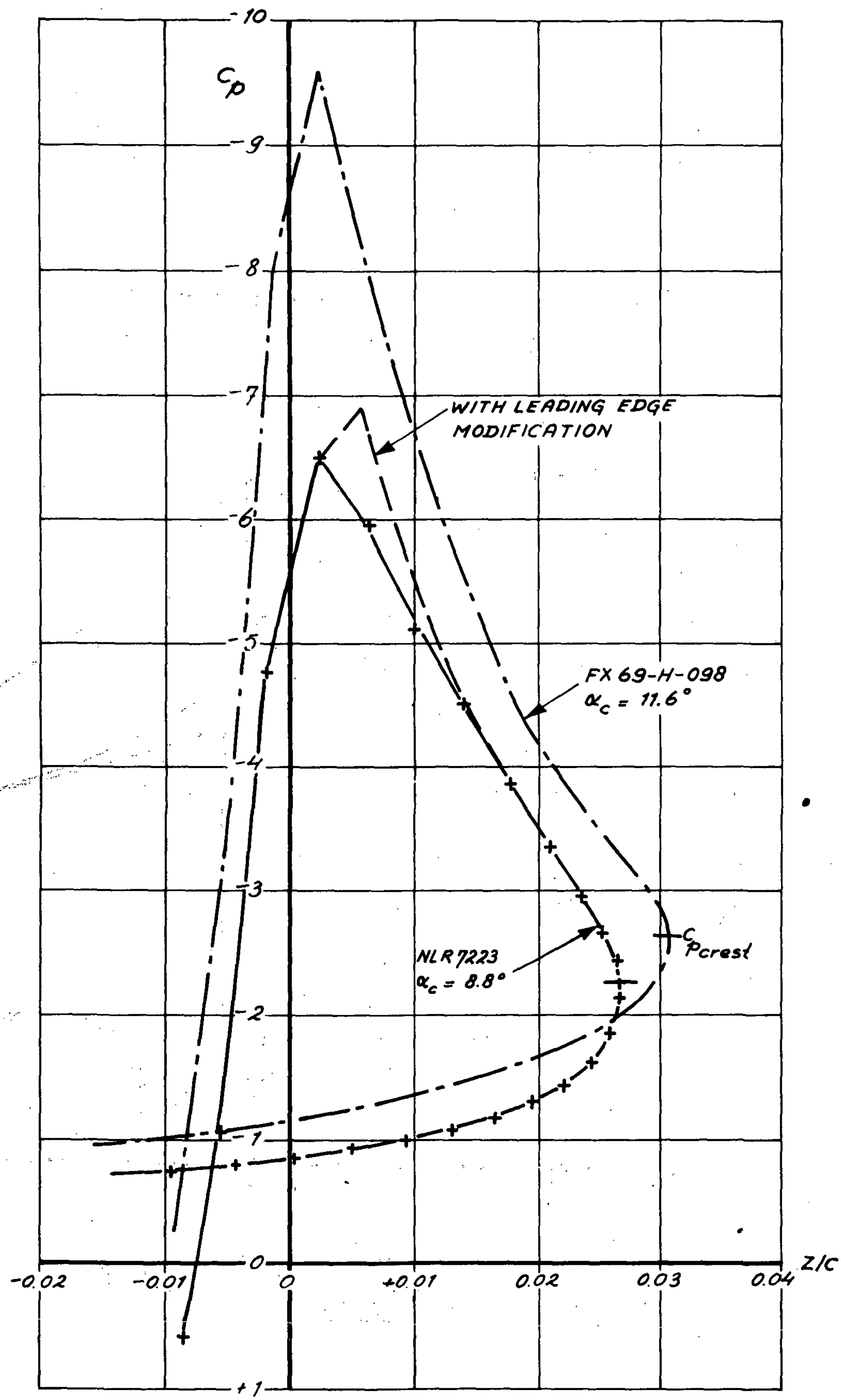


FIG. 9 AIRFOIL 2.
SUCTION LOOP FOR OPTIMUM ANGLE OF ATTACK AT $Ma = 0$.

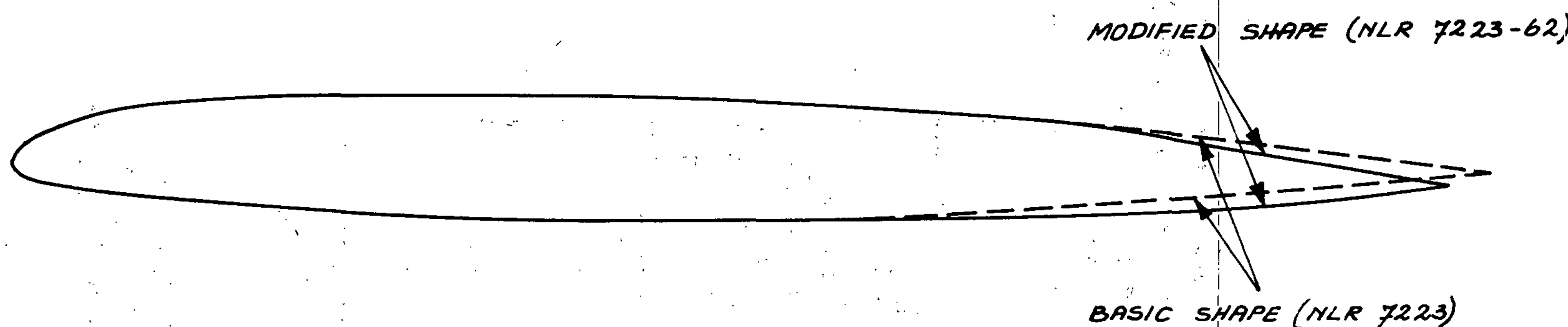
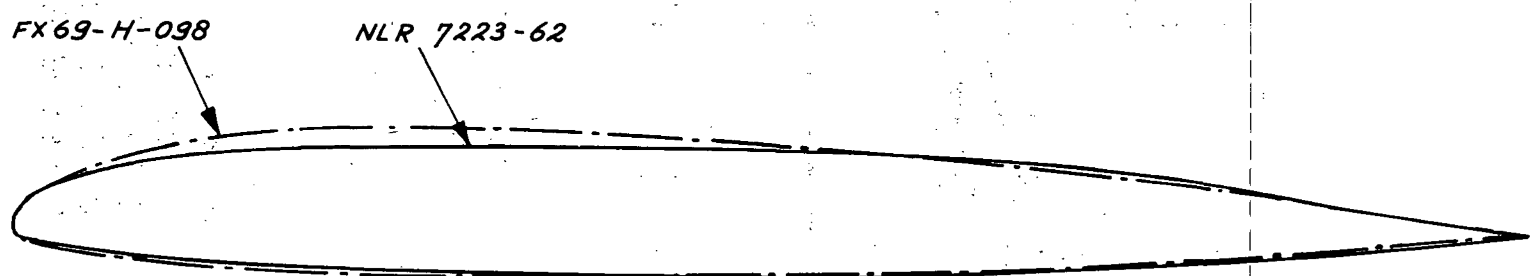


FIG. 10 AIRFOIL 1.
COMPARISON OF BASIC AND MODIFIED SHAPES.



	$(t/c)_{max}$	$(x/c)_{t_{max}}$	$(R/c)_{nose}$	$\Theta_{t.e.}$	α_o (Ma = 0)	c_{m_o} (Ma = 0)
FX69-H-098	0.098	0.30	0.0062	13.2°	-0.8°	-0.010
NLR 7223-62	0.086	0.38	0.0060	12.6°	-0.9°	-0.011

FIG. 11 AIRFOIL 1.
COMPARISON WITH FX69-H-098 AIRFOIL SHAPE.

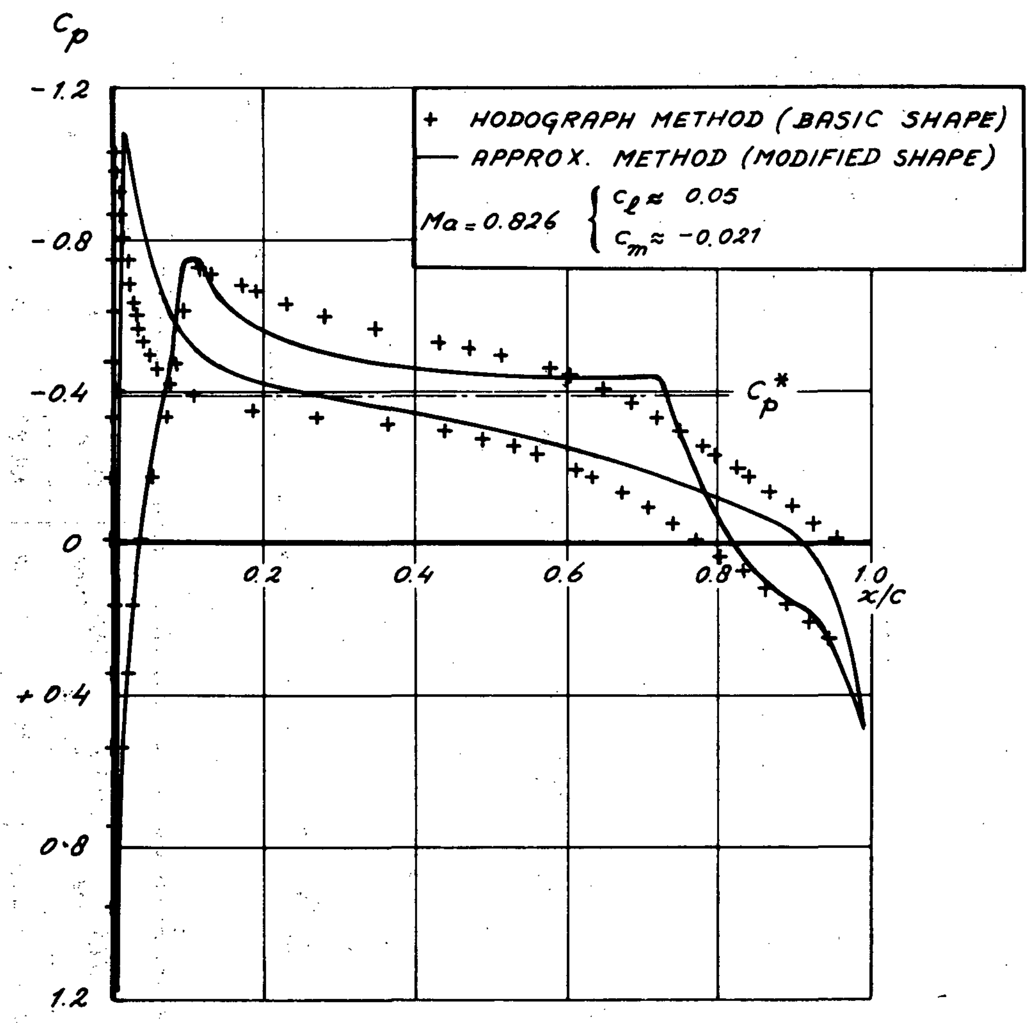


FIG. 12 AIRFOIL 1.
COMPARISON OF APPROXIMATE POTENTIAL FLOW PRESSURE DISTRIBUTION WITH BASIC HODOGRAPH SOLUTION FOR HIGH SPEED, LOW c_ℓ-DESIGN CONDITION.

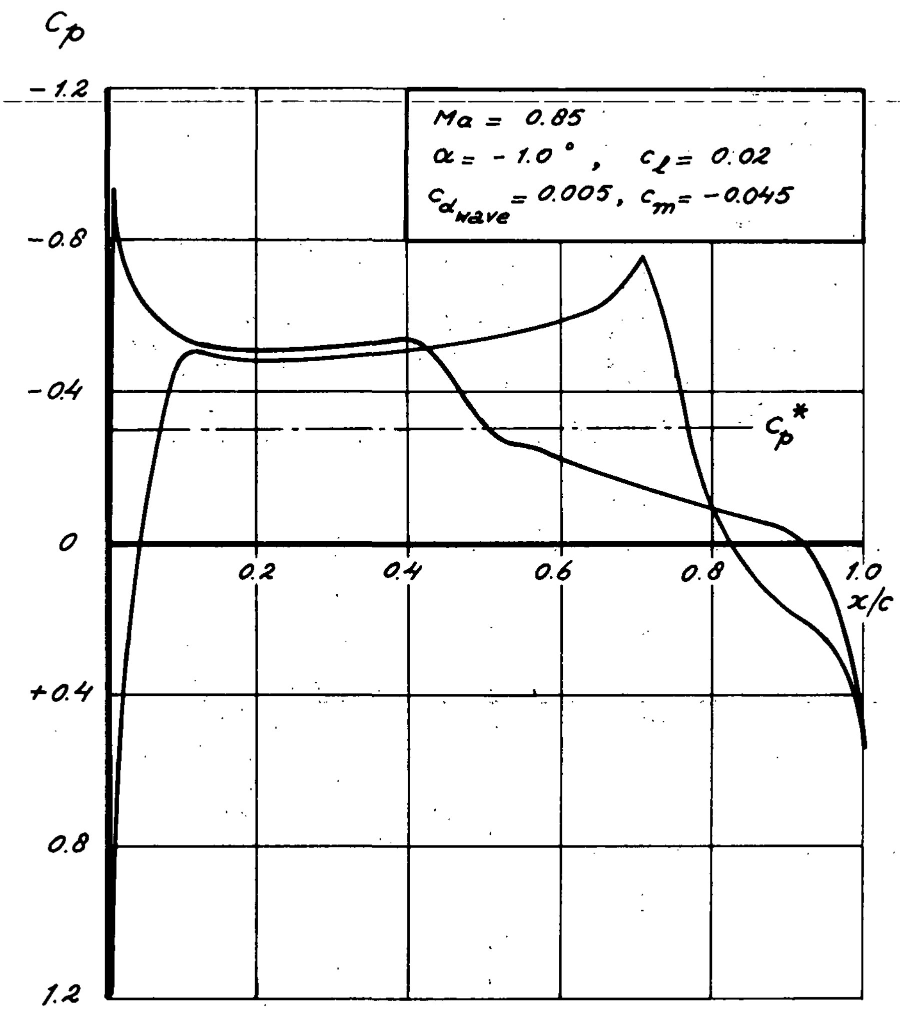


FIG. 13 AIRFOIL 1.
PRESSURE DISTRIBUTION FOR INVISCID FLOW AT $Ma = 0.85$, $c_\ell \approx 0$ CALCULATED WITH THE GARABEDIAN/KORN RELAXATION METHOD. (CRUDE MESH).

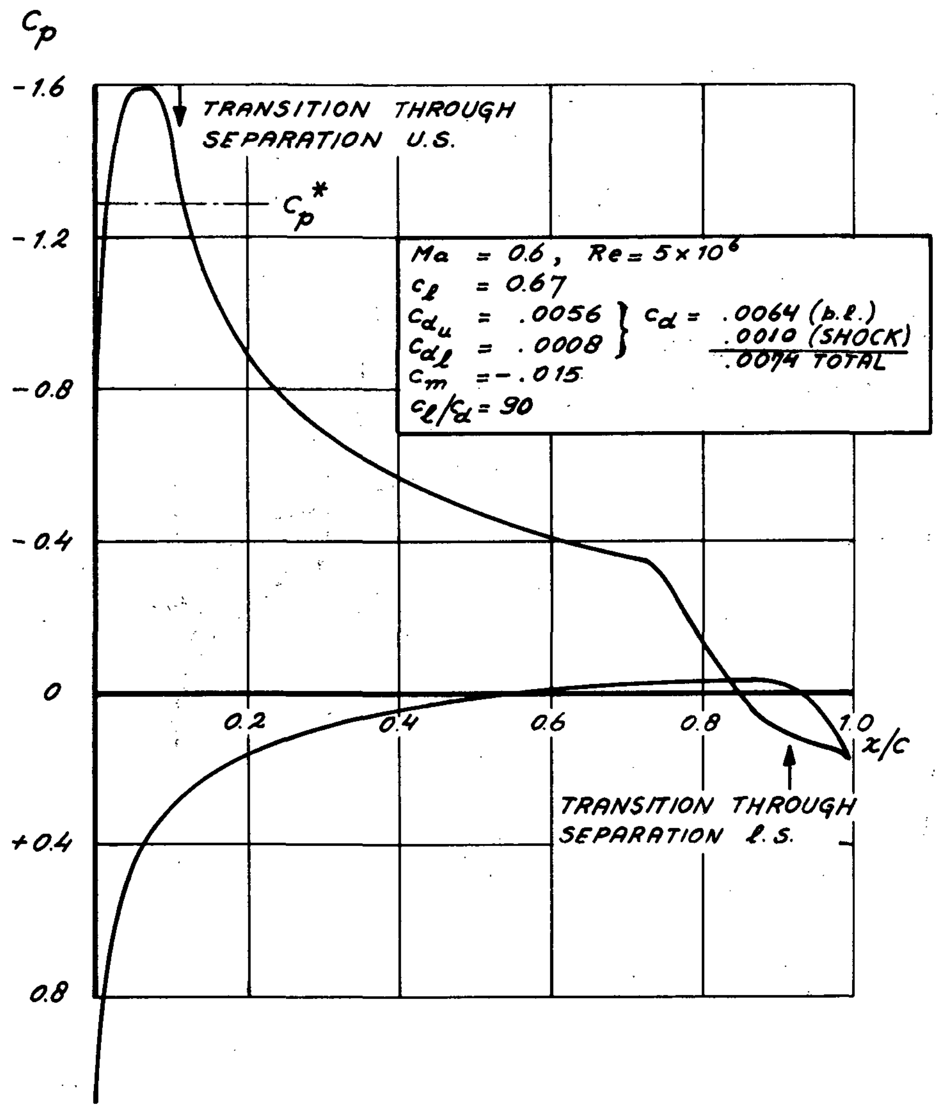


FIG. 14 AIRFOIL 1.
CALCULATED PRESSURE DISTRIBUTION (APPROXIMATE METHOD, INCLUDING EFFECT OF THE BOUNDARY LAYER) AT THE HOVER CONDITION.

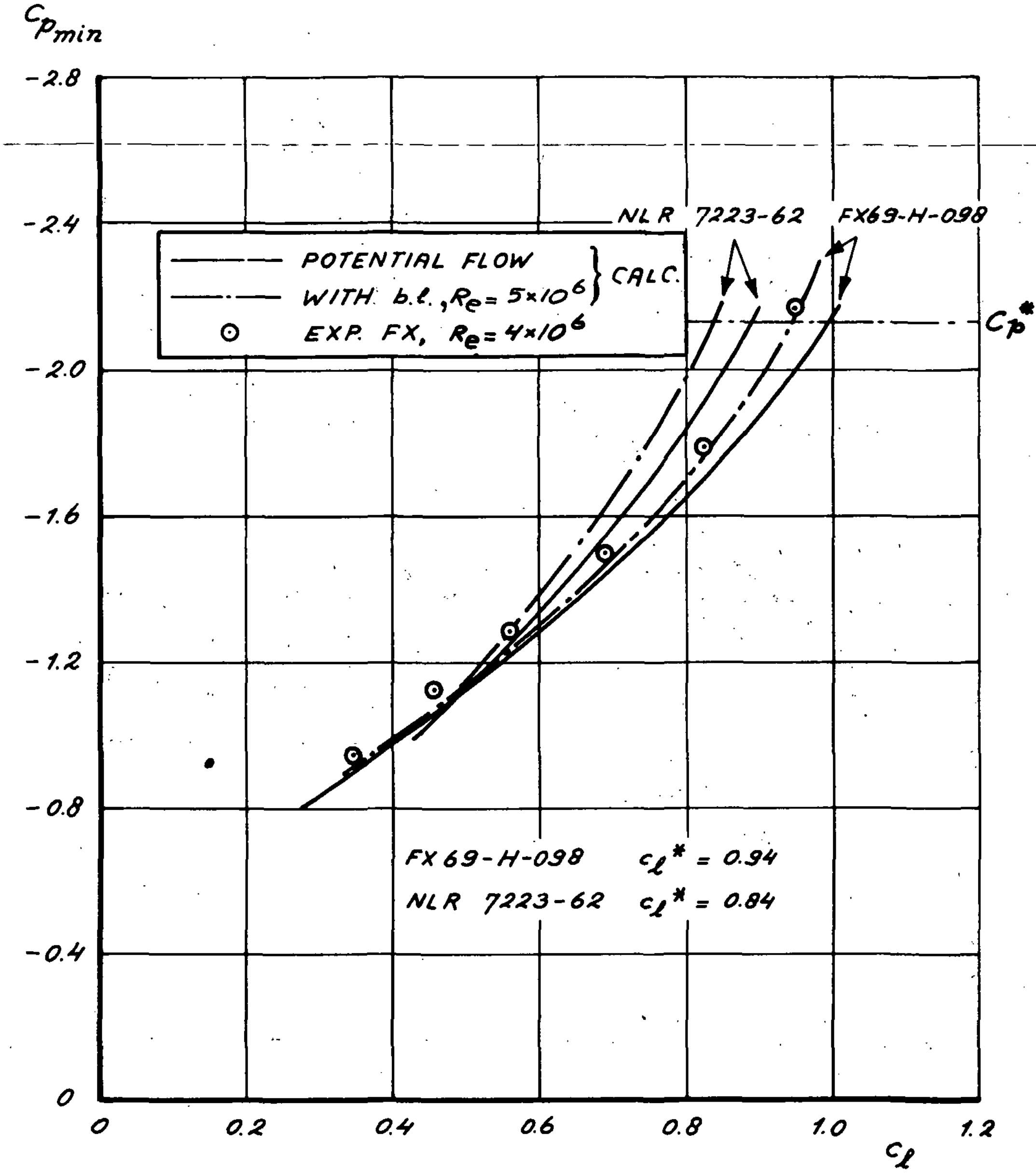


FIG. 15 AIRFOIL 1.
MINIMUM PRESSURE AS A FUNCTION OF c_ℓ AT $Ma = 0.5$.

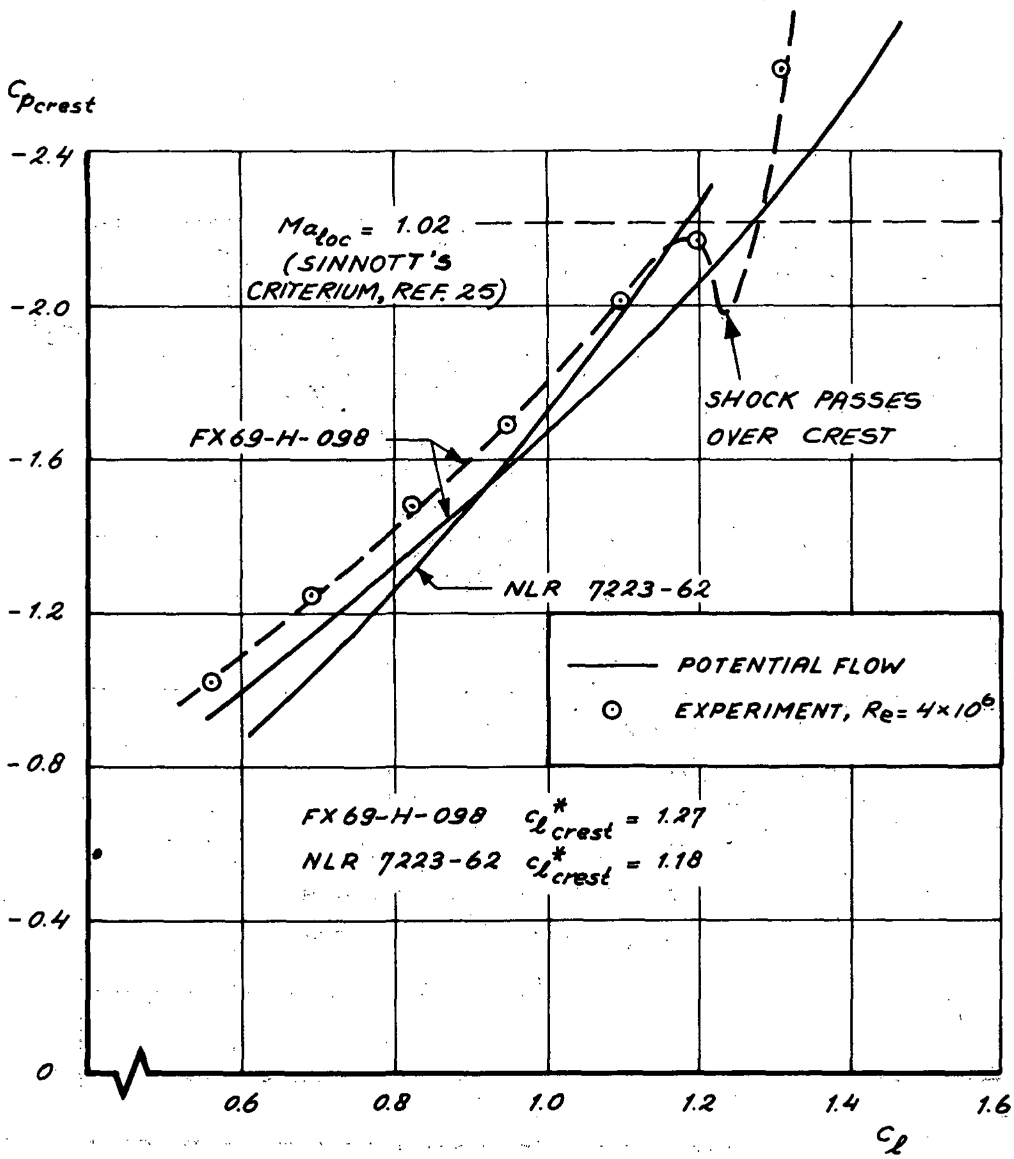


FIG. 16 AIRFOIL 1.
CREST PRESSURE AS A FUNCTION
OF c_ℓ AT Ma = 0.5.

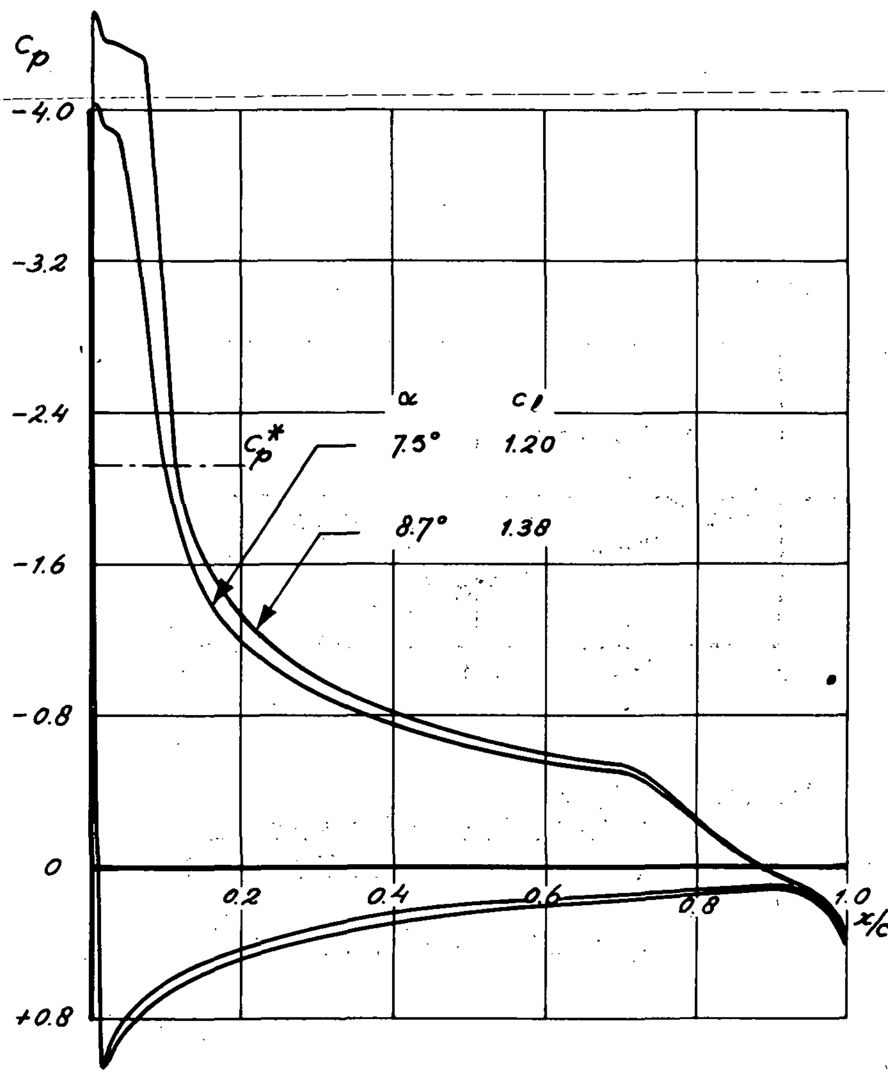


FIG. 17 AIRFOIL 1.
PRESSURE DISTRIBUTIONS FOR INVISCID, SUPERCRITICAL FLOW AT Ma = 0.5 AS CALCULATED BY MEANS OF THE GARABEDIAN/KORN RELAXATION METHOD (CRUDE MESH).

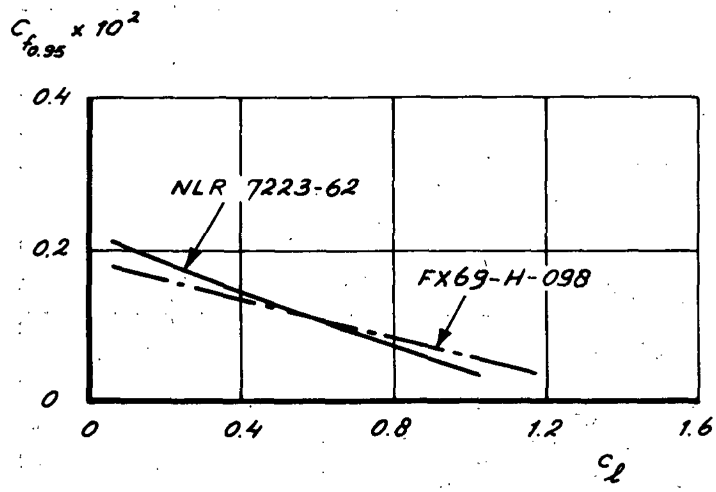


FIG. 18 AIRFOIL 1.
SKIN FRICTION COEFFICIENT AT $x/c = 0.95$ AS A FUNCTION OF c_ℓ AT $Ma = 0.5$, $Re = 5 \times 10^6$ (NASH LOCAL EQUILIBRIUM METHOD).

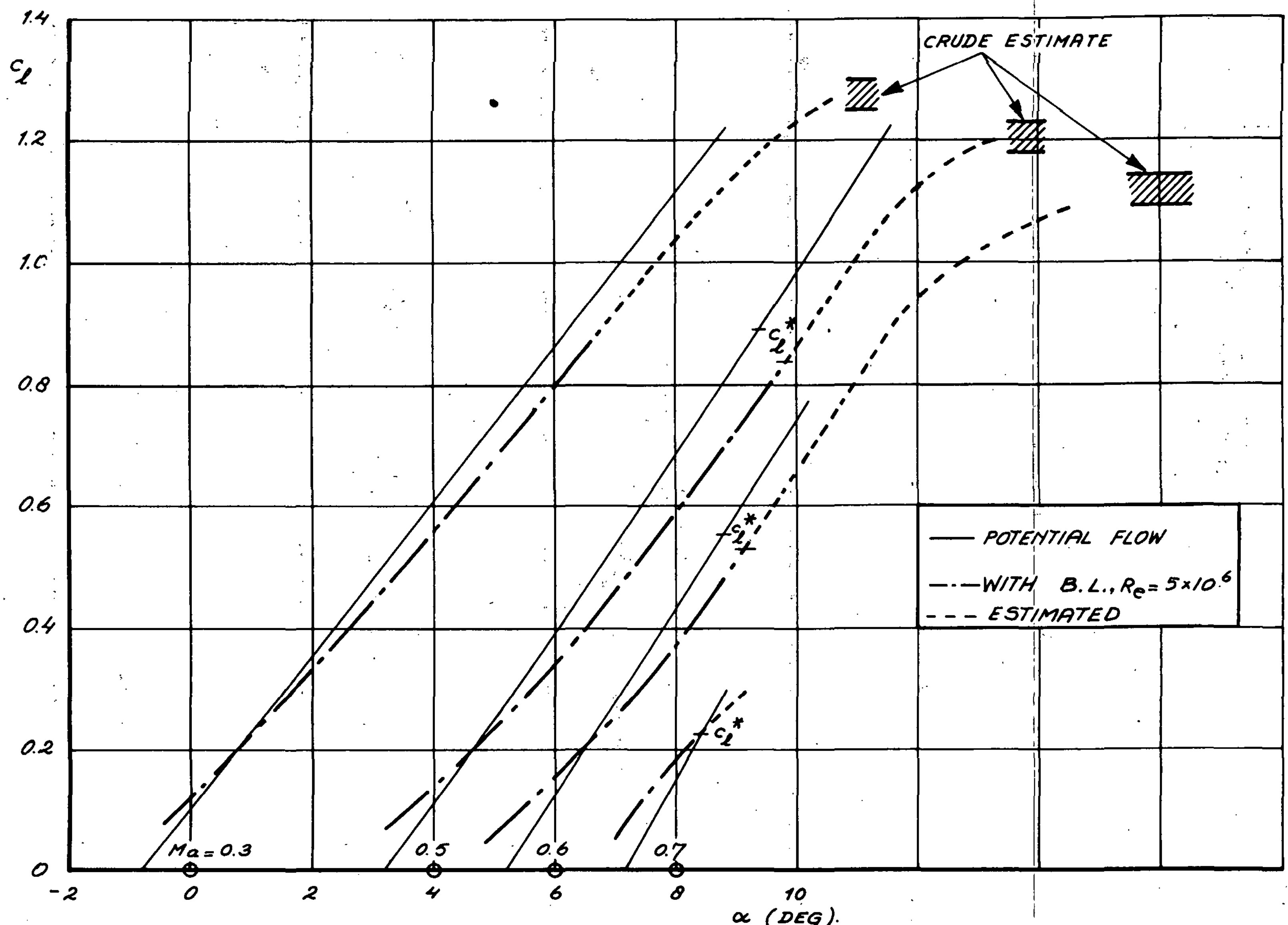


FIG. 19. AIRFOIL 1.
c_ℓ VERSUS α FOR VARIOUS MACH NUMBERS (ESTIMATED)

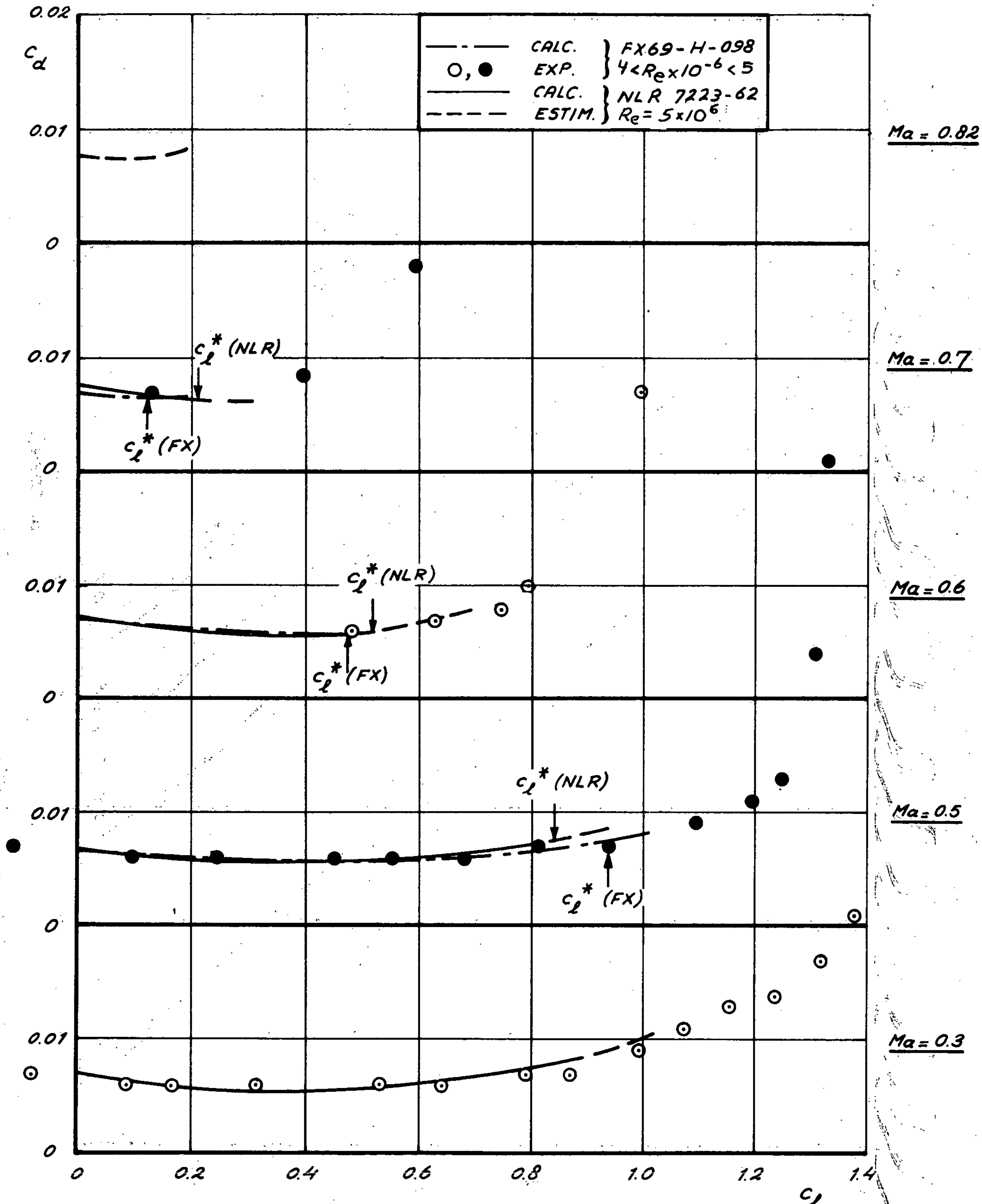


FIG. 20 AIRFOIL 1.
ESTIMATED DRAG POLARS FOR VARIOUS MACH NUMBERS.

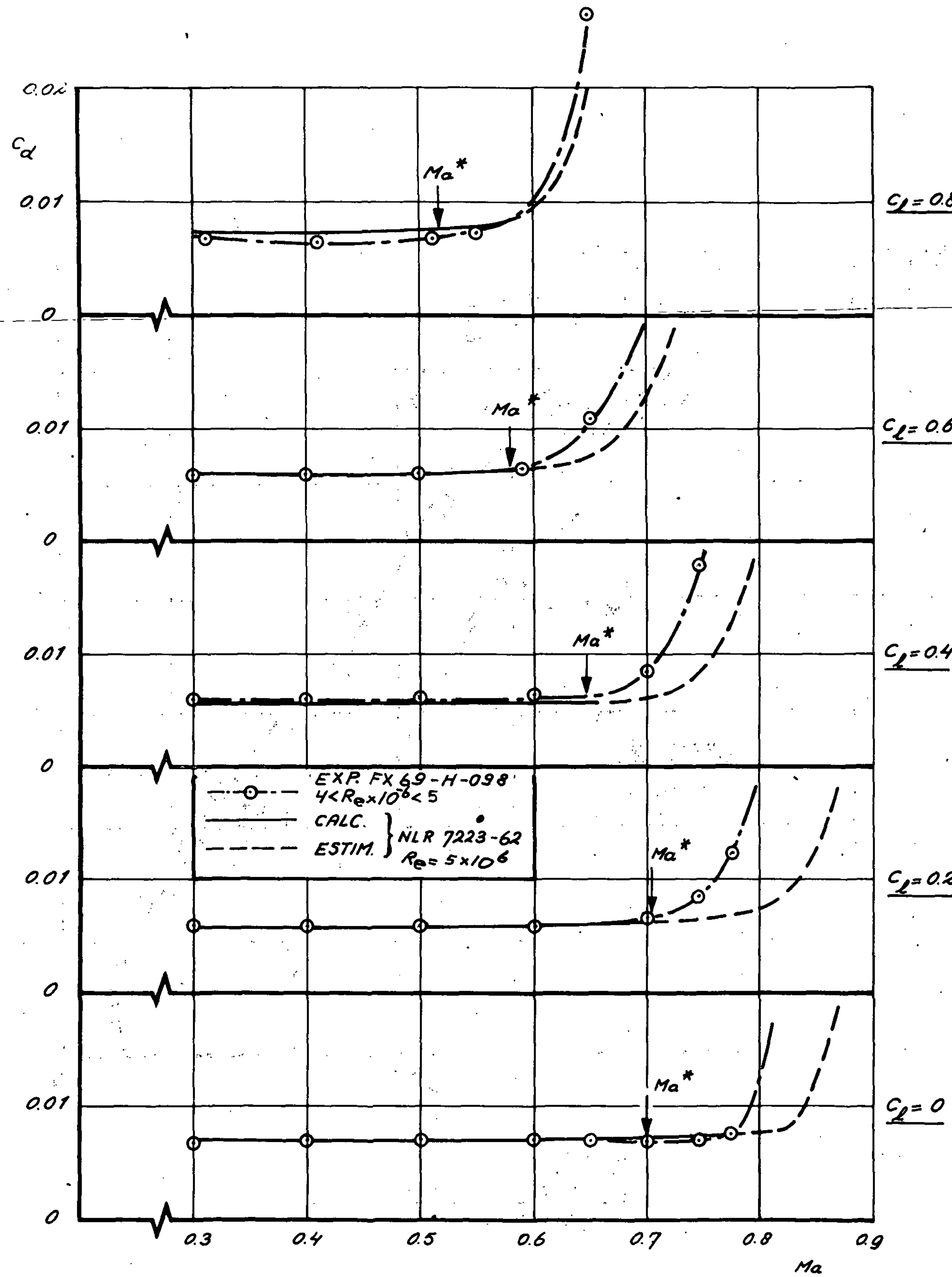


FIG. 21 AIRFOIL 1.
ESTIMATED DRAG AT CONSTANT LIFT FOR VARIOUS MACH NUMBERS.

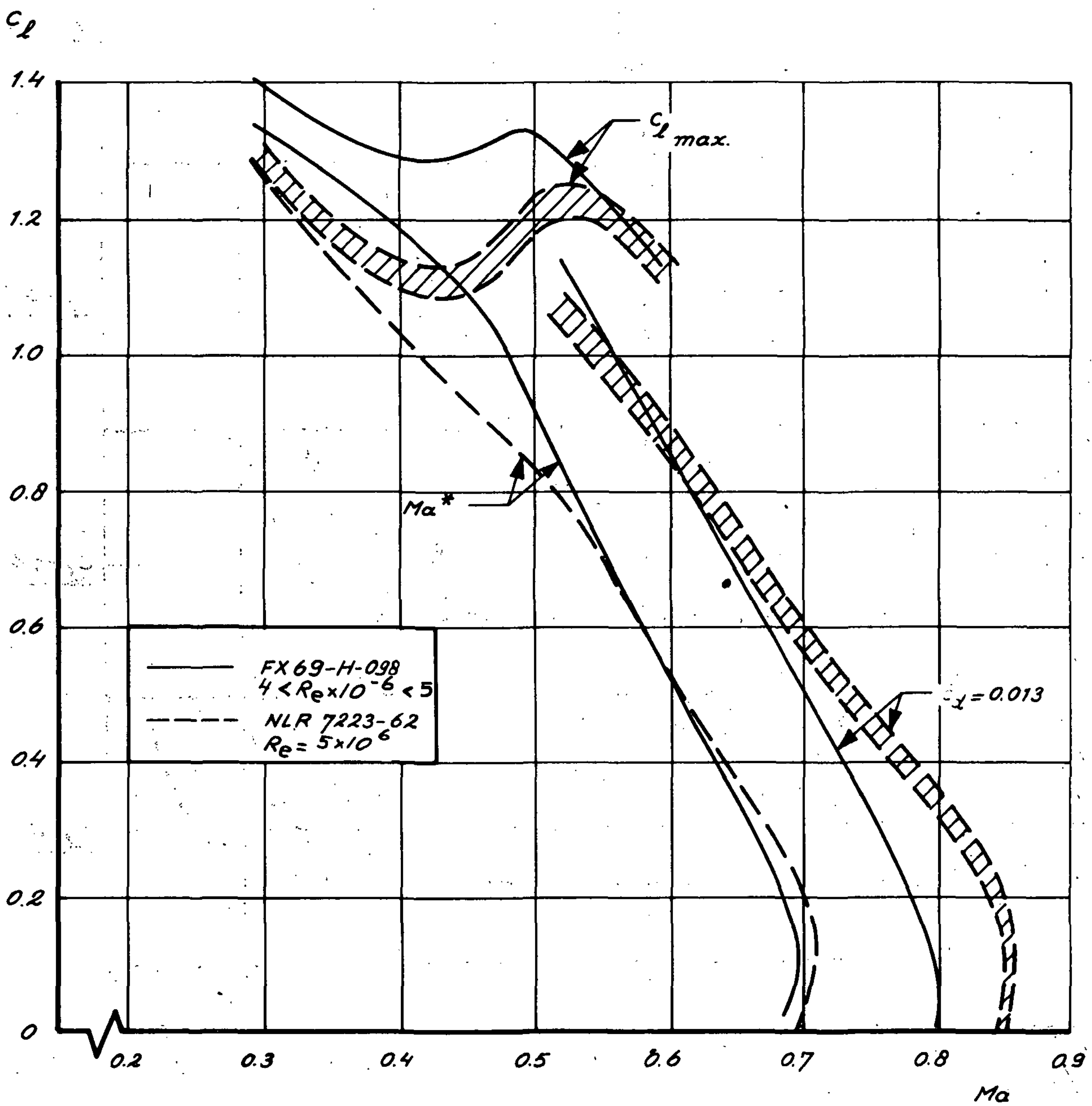


FIG. 22 AIRFOIL 1.
ESTIMATED BOUNDARIES IN
c_ℓ - Ma PLANE.

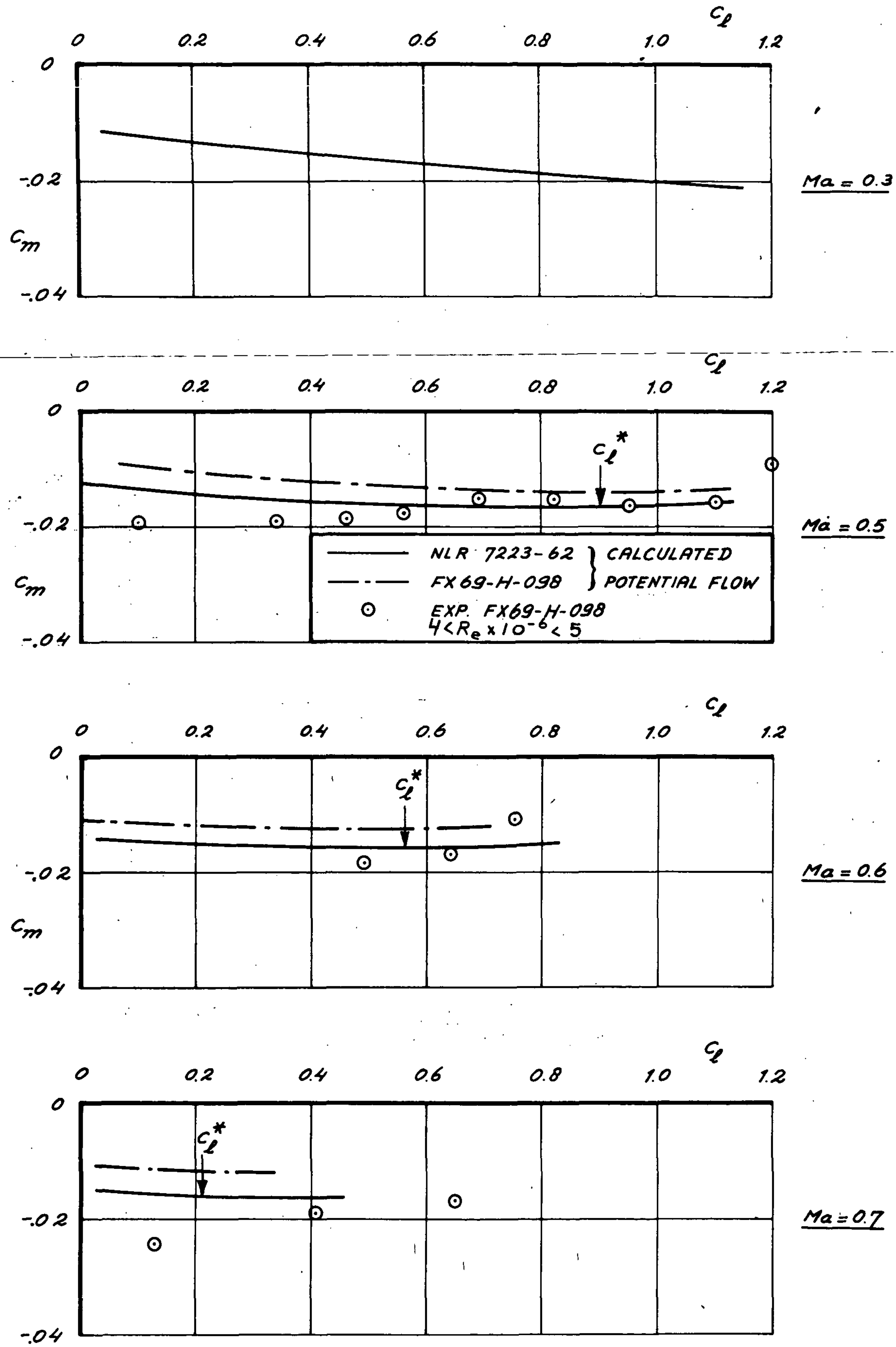


FIG. 23 AIRFOIL 1.
PITCHING MOMENT AS A FUNCTION OF c_ℓ
FOR VARIOUS MACH NUMBERS.

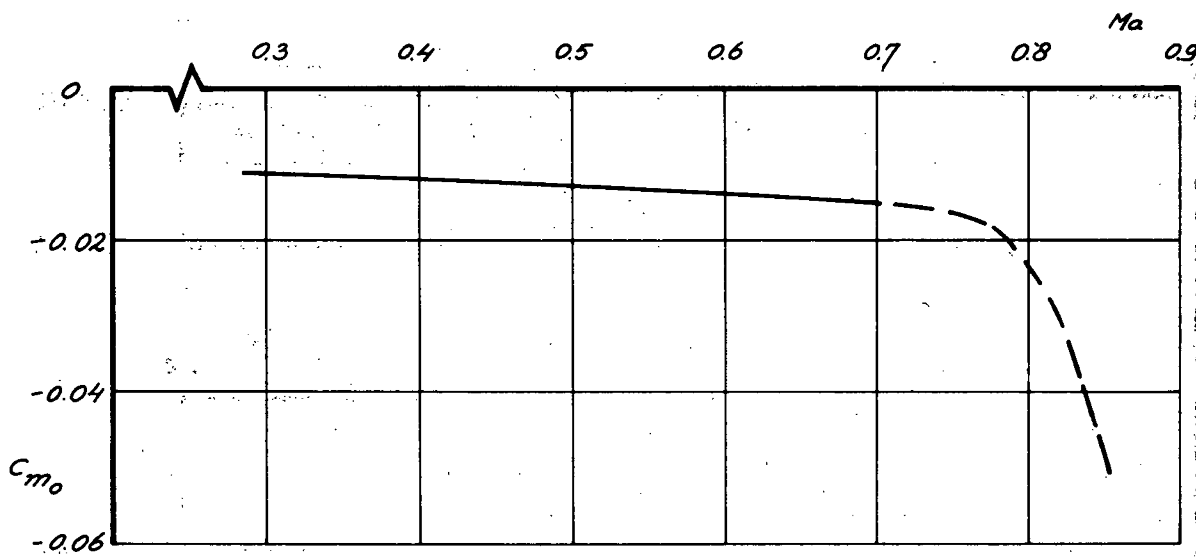


FIG. 24 AIRFOIL 1.
ZERO LIFT PITCHING MOMENT AS A FUNCTION OF MACH NUMBER (INVISCID FLOW).

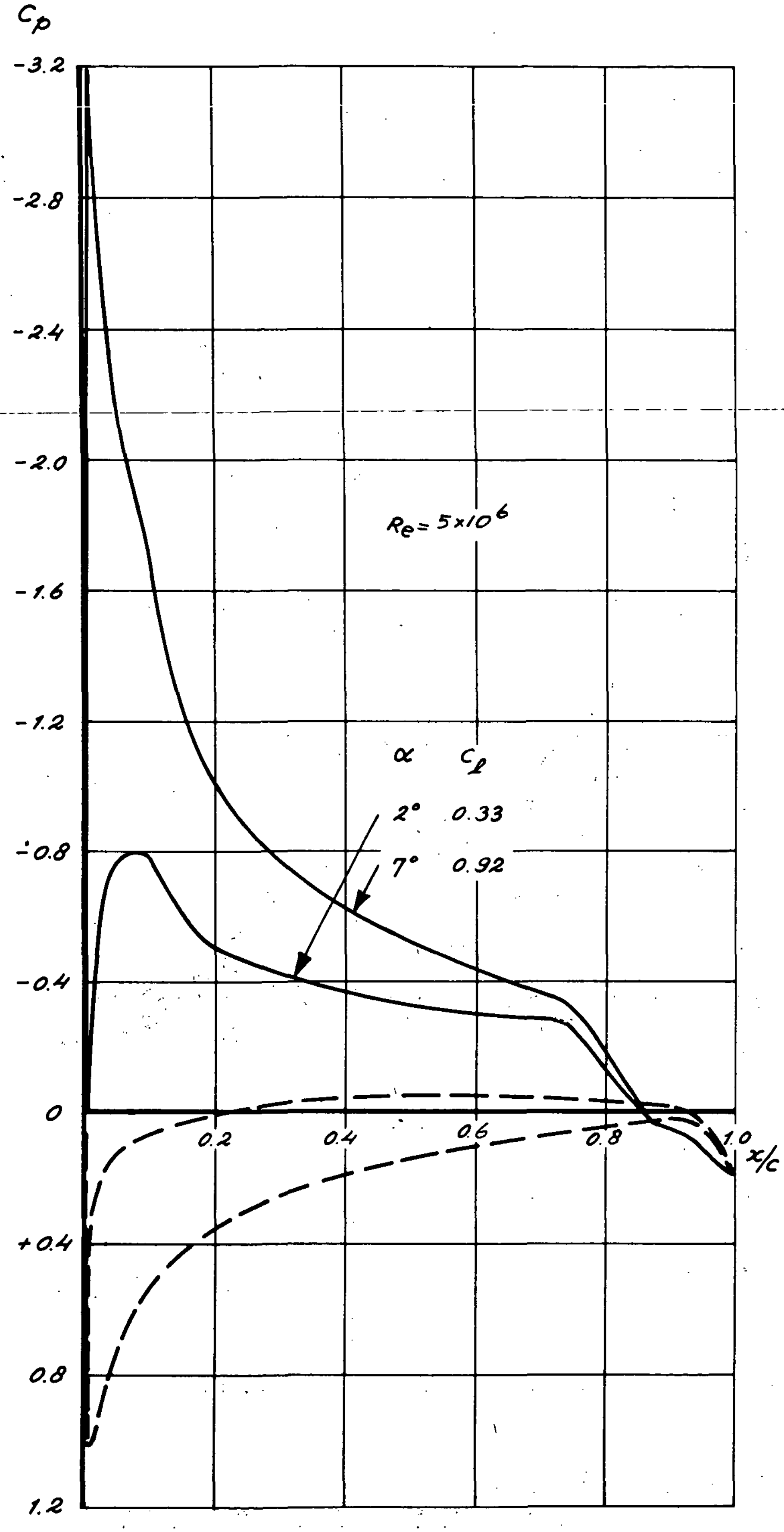


FIG. 25 AIRFOIL 1.
PRESSURE DISTRIBUTIONS AT Ma = 03, INCLUDING EFFECT OF THE BOUNDARY LAYER (APPROX. METHOD).

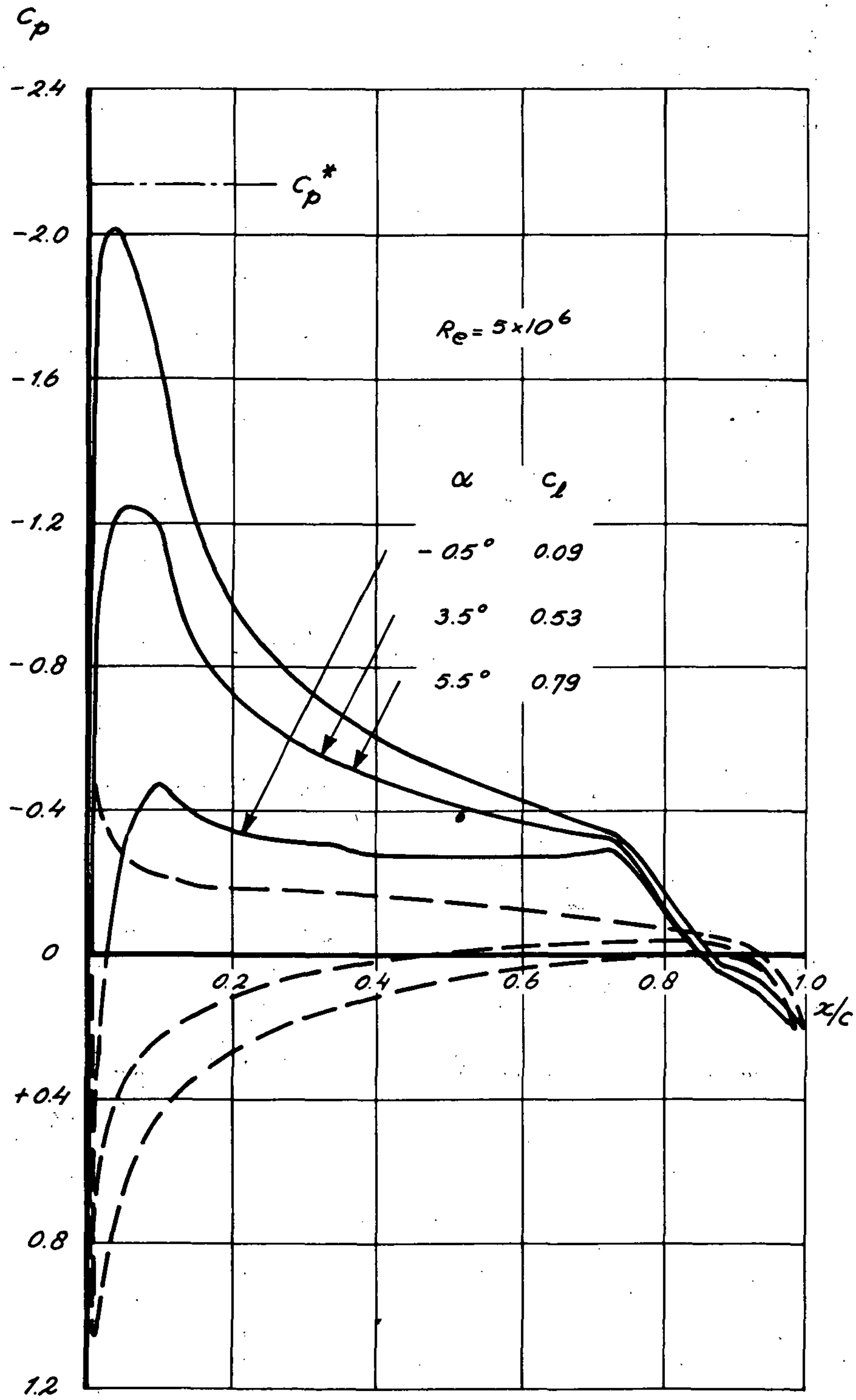


FIG. 26 AIRFOIL 1.
PRESSURE DISTRIBUTIONS AT $Ma = 0.5$,
INCLUDING EFFECT OF THE BOUNDARY
LAYER (APPROX. METHOD).

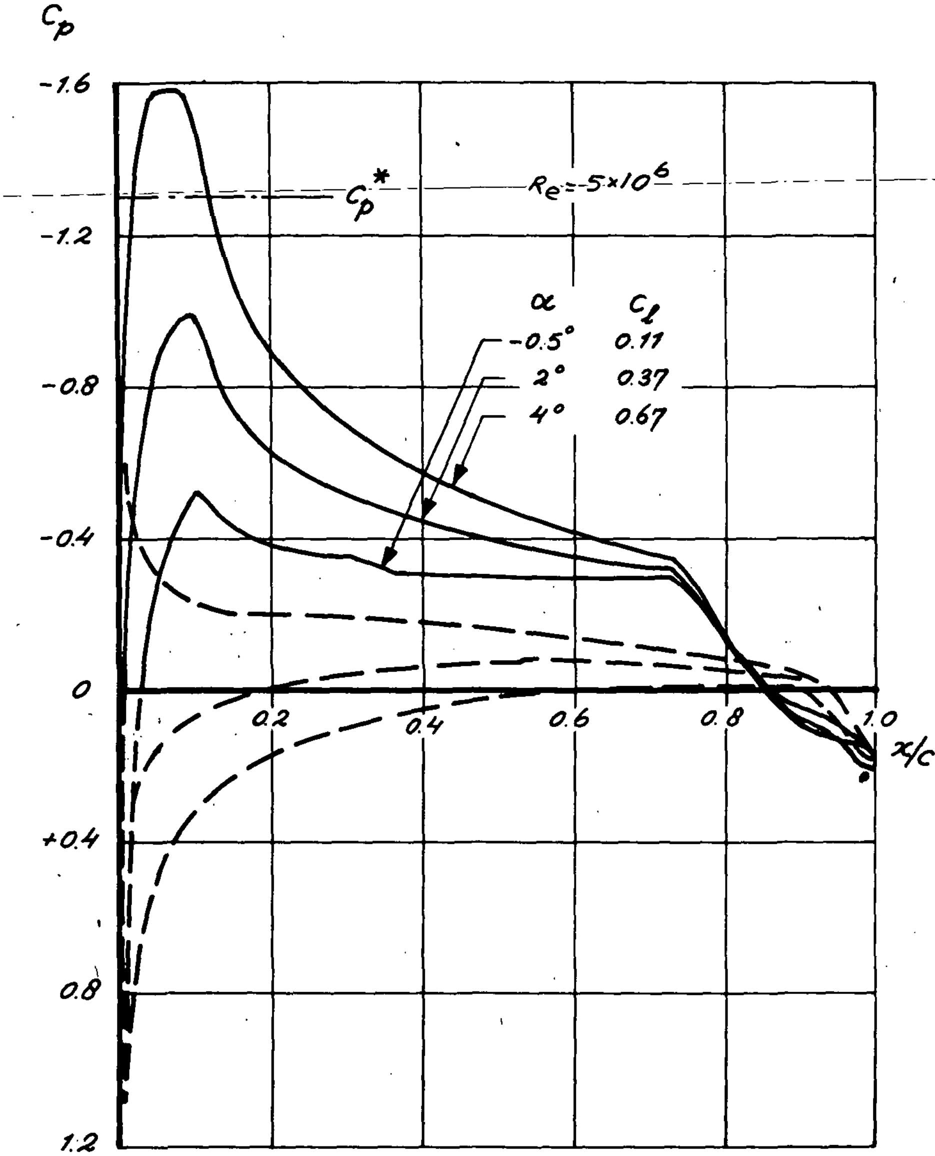


FIG. 27 AIRFOIL 1.
PRESSURE DISTRIBUTIONS AT Ma = 0.6, INCLUDING EFFECT OF THE BOUNDARY LAYER (APPROX. METHOD).

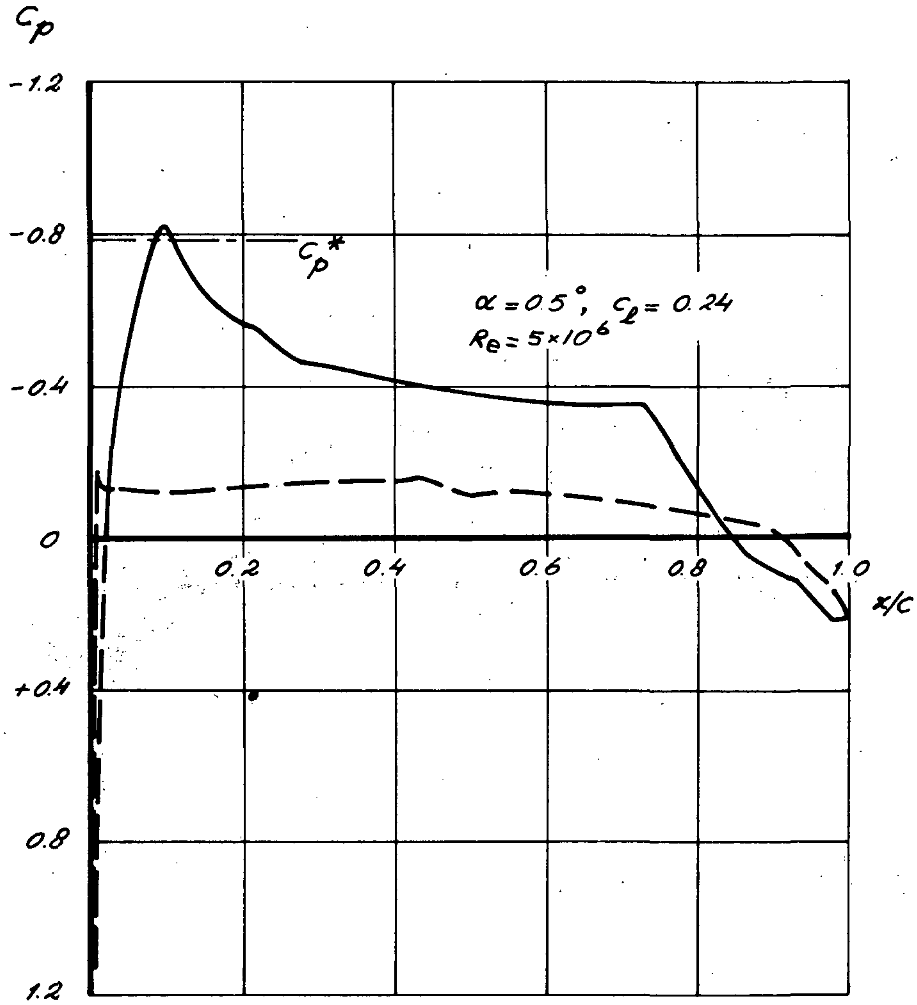


FIG 28 AIRFOIL 1.
PRESSURE DISTRIBUTION AT Ma = 0.7, INCLUDING EFFECT OF THE BOUNDARY LAYER (APPROX. METHOD).

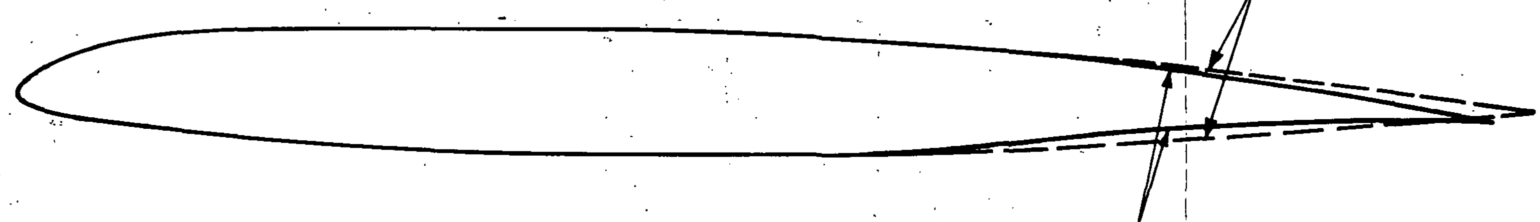
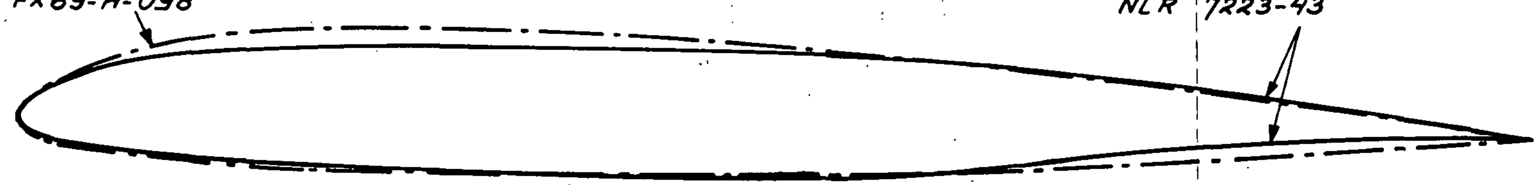


FIG. 29 AIRFOIL 2.
COMPARISON OF BASIC AND MODIFIED SHAPES.



	$(t/c)_{max}$	$(x/c)_{t_{max}}$	$(R/c)_{nose}$	$\Theta_{t.e.}$	α_o (Ma = 0)	c_{m_o} (Ma = 0)
FX69-H-098	0.098	0.30	0.0062	13.2°	-0.8°	-0.010
NLR 7223-43	0.086	0.38	0.0060	9.3°	-2.2°	-0.051

FIG. 30 AIRFOIL 2.
COMPARISON WITH FX69-H-098 AIRFOIL SHAPE.

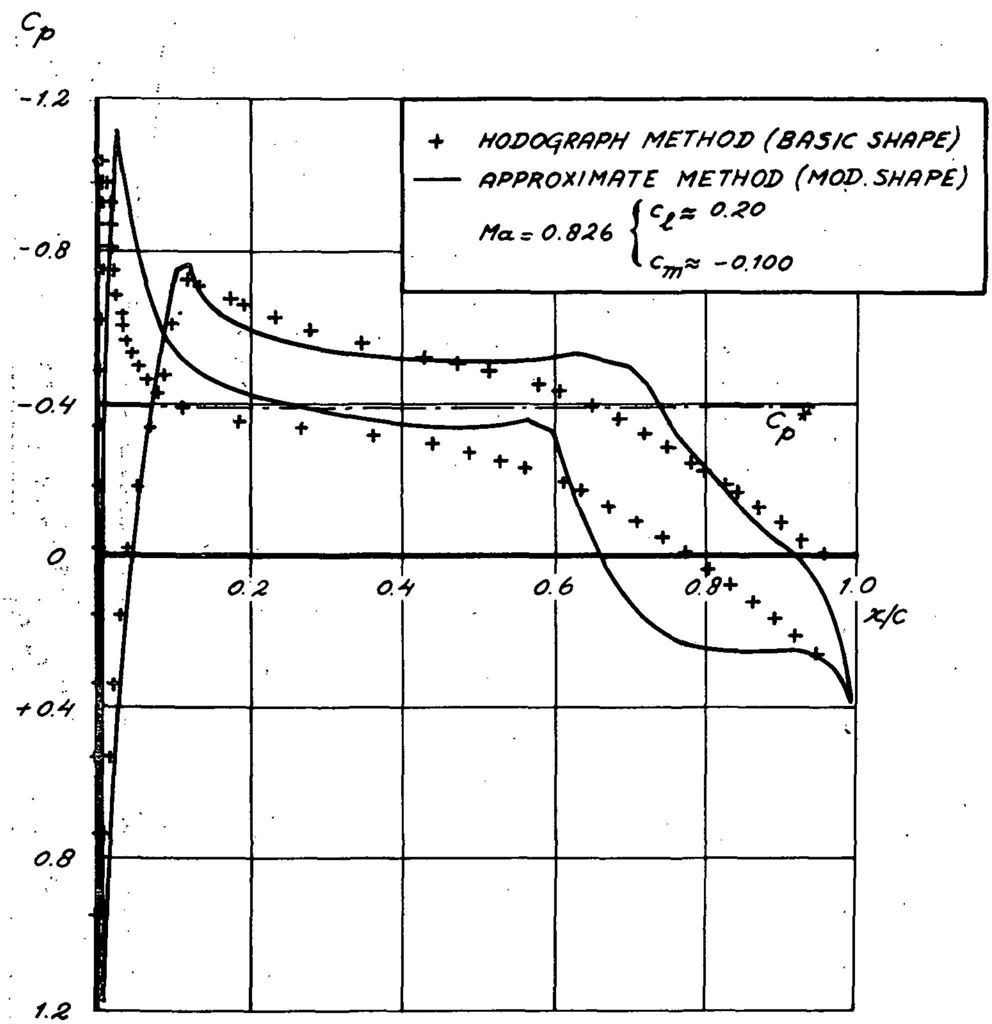


FIG. 31 AIRFOIL 2.
COMPARISON OF APPROXIMATE POTENTIAL FLOW PRESSURE DISTRIBUTION WITH BASIC HODOGRAPH SOLUTION FOR HIGH SPEED, LOW c_ℓ DESIGN CONDITION.

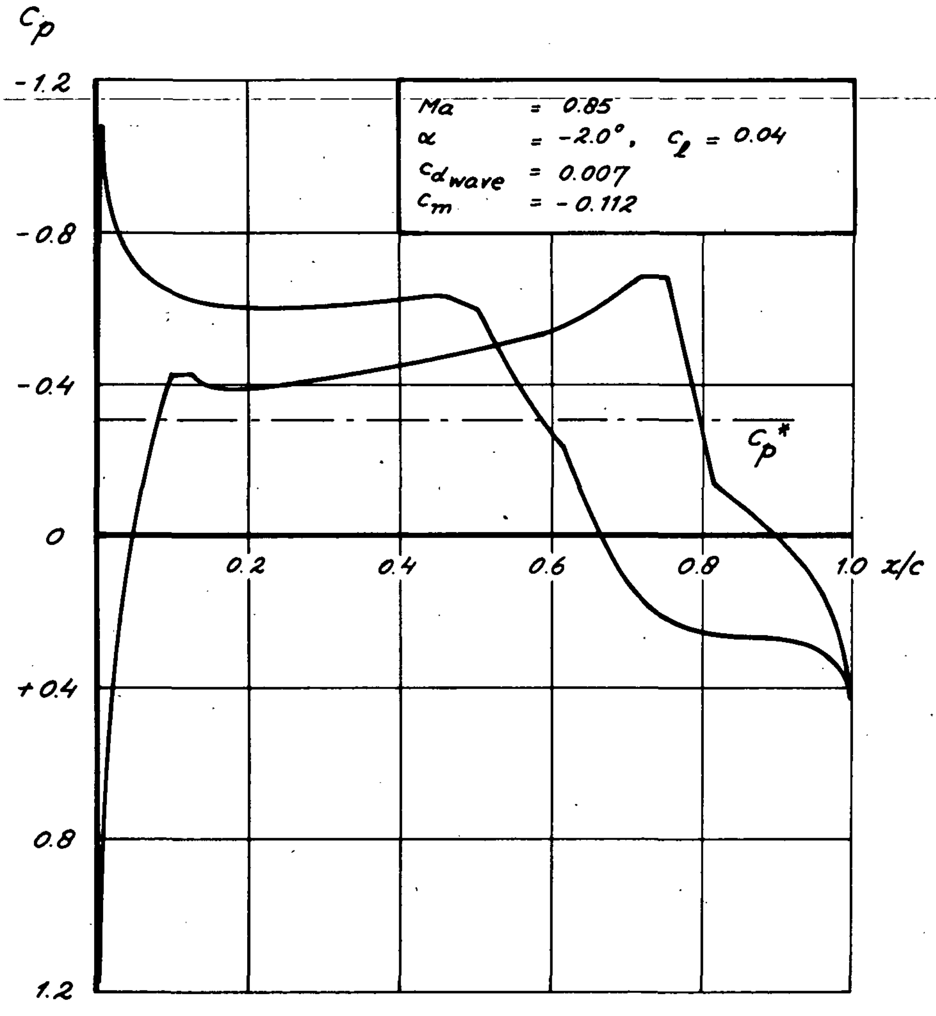


FIG. 32 AIRFOIL 2.
PRESSURE DISTRIBUTION FOR INVISCID FLOW AT $Ma = 0.85$, $c_l \approx 0$ CALCULATED BY MEANS OF THE GARABEDIAN/KORN RELAXATION METHOD (CRUDE MESH).

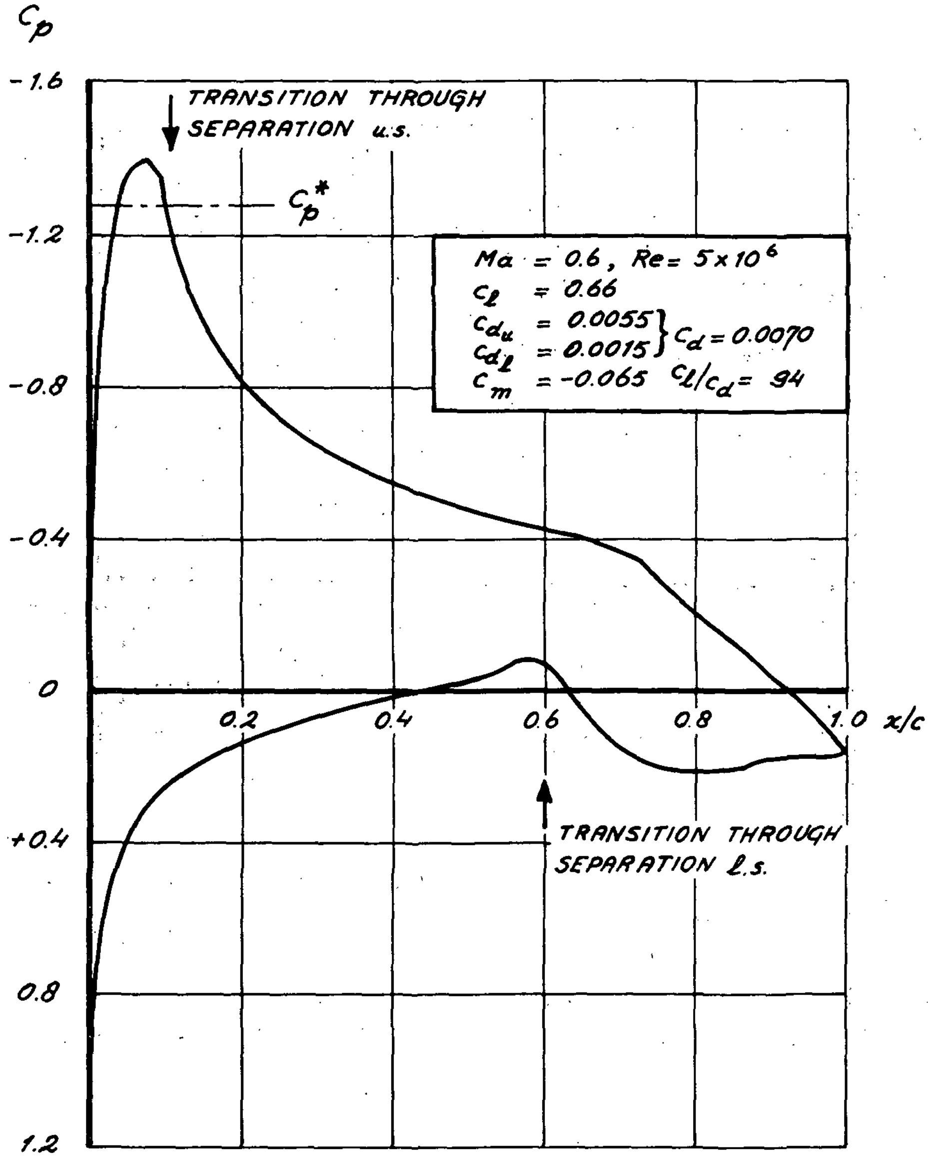


FIG. 33. AIRFOIL 2. CALCULATED PRESSURE DISTRIBUTION (APPROXIMATE METHOD INCLUDING EFFECT OF BOUNDARY LAYER) AT THE HOVER CONDITION.

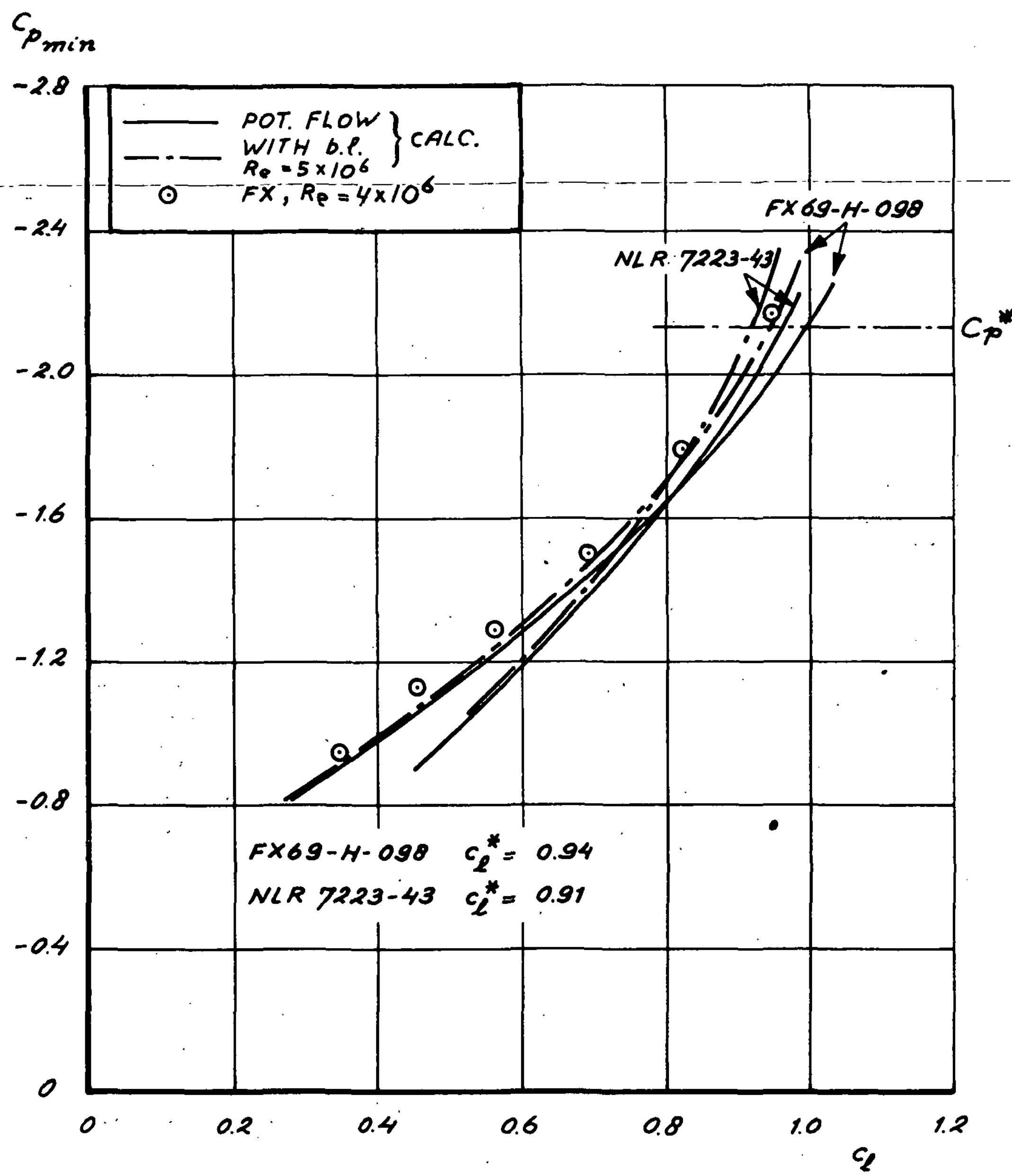


FIG. 34 AIRFOIL 2.
MINIMUM PRESSURE AS A FUNCTION OF c_l AT $Ma = 0.5$.

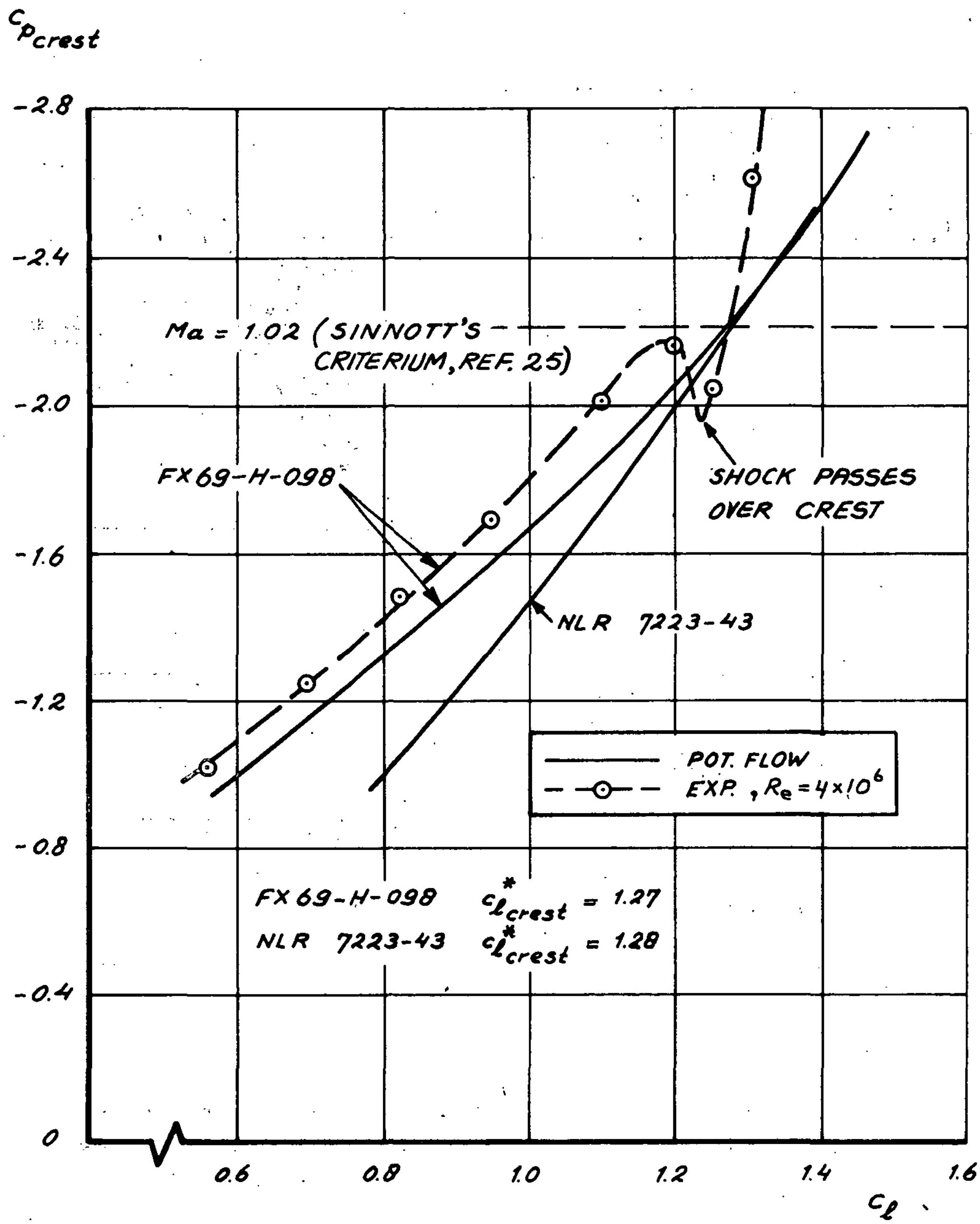


FIG. 35 AIRFOIL 2.
CREST PRESSURE AS A FUNCTION OF c_l.
AT MACH = 0.5

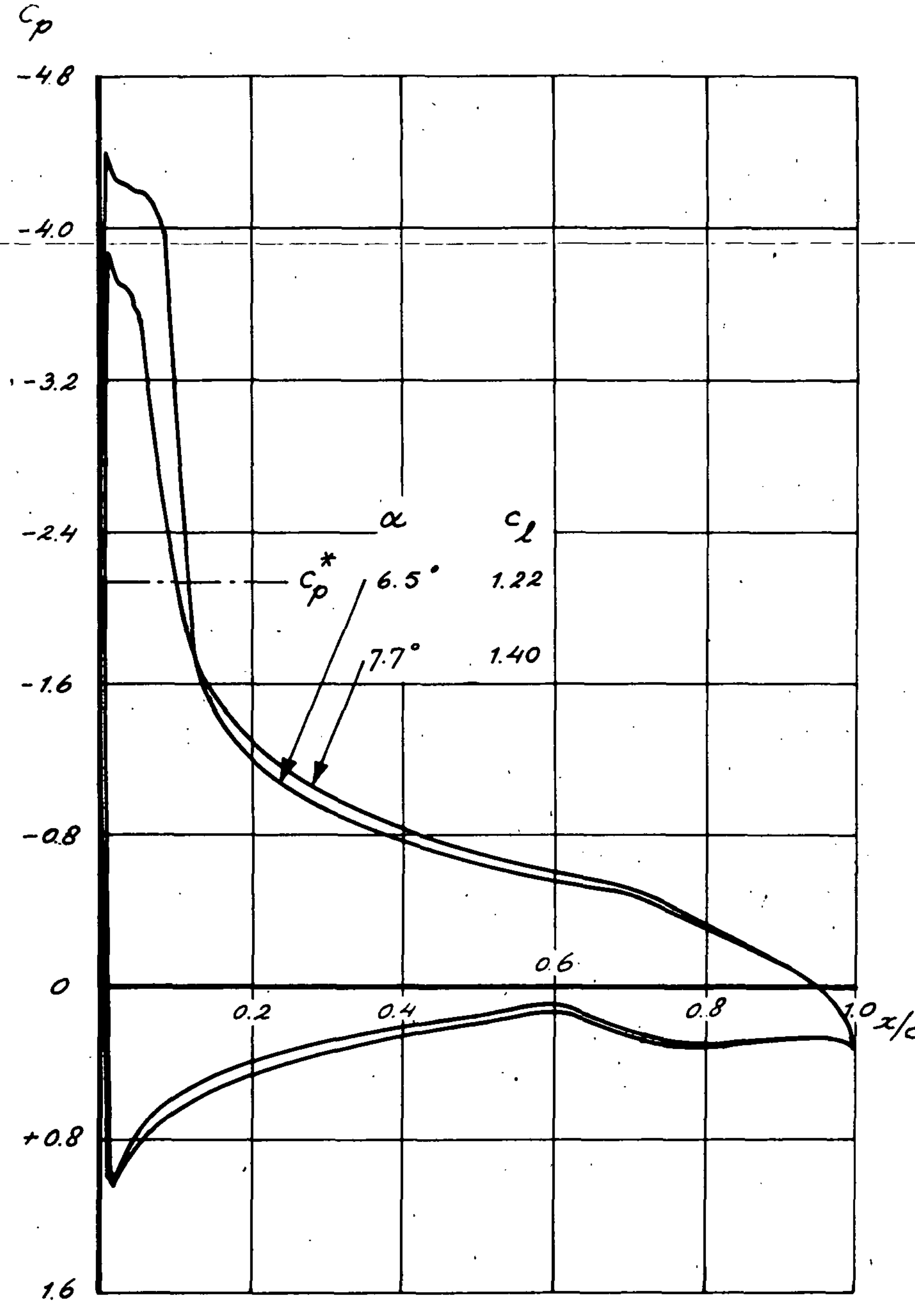


FIG. 36 AIRFOIL 2.
PRESSURE DISTRIBUTION FOR INVISCID, SUPERCRITICAL FLOW AT $M_\alpha = 0.5$ AS CALCULATED BY MEANS OF THE GARABEDIAN/KORN RELAXATION METHOD (CRUDE MESH).

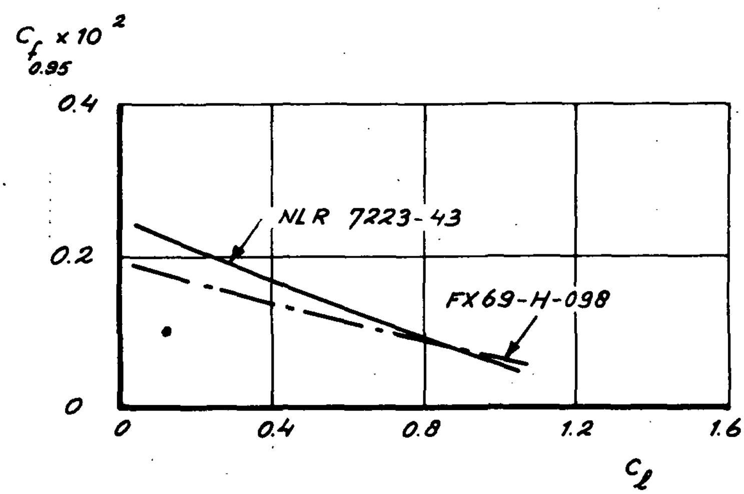


FIG. 37 AIRFOIL 2.
SKIN FRICTION COEFFICIENT AT $x/c = 0.95$ AS A FUNCTION OF c_l AT $Ma = 0.5$, $Re = 5 \times 10^6$ (NASH LOCAL EQUILIBRIUM METHOD).

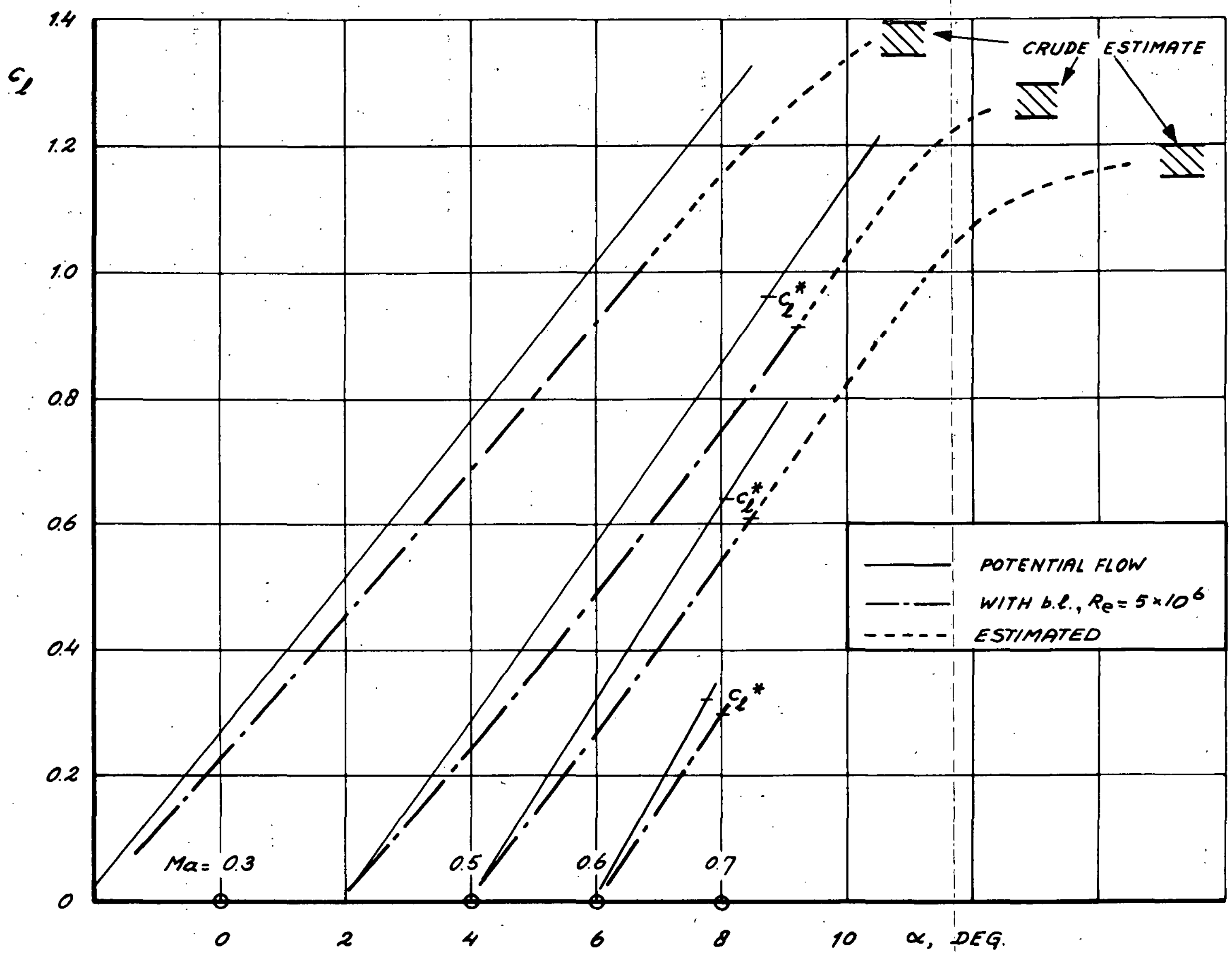


FIG. 38 AIRFOIL 2.
c_l VERSUS α FOR SEVERAL MACH NUMBERS (ESTIMATED)

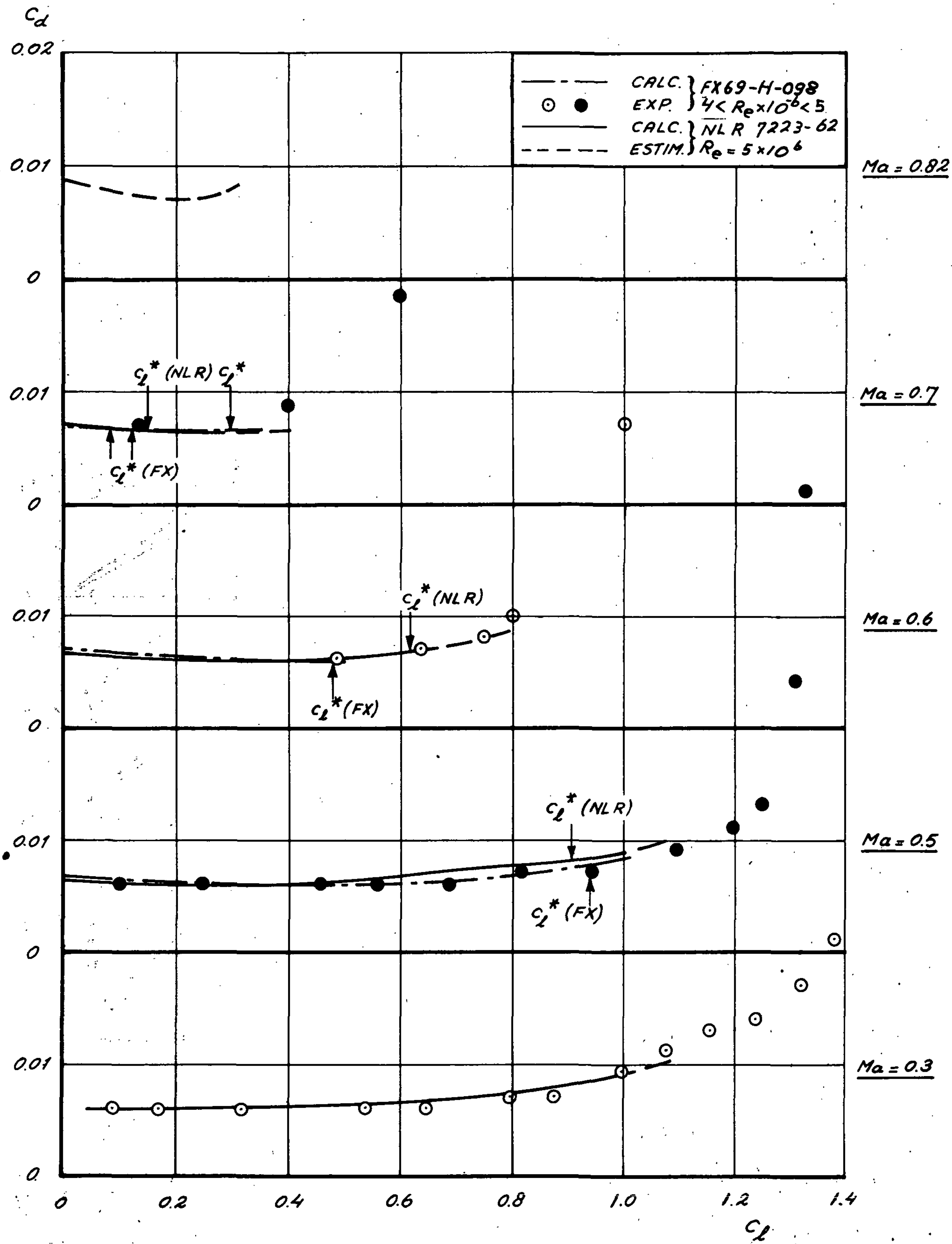


FIG. 39 AIRFOIL 2.
ESTIMATED DRAG POLARS FOR SEVERAL MACH NUMBERS.

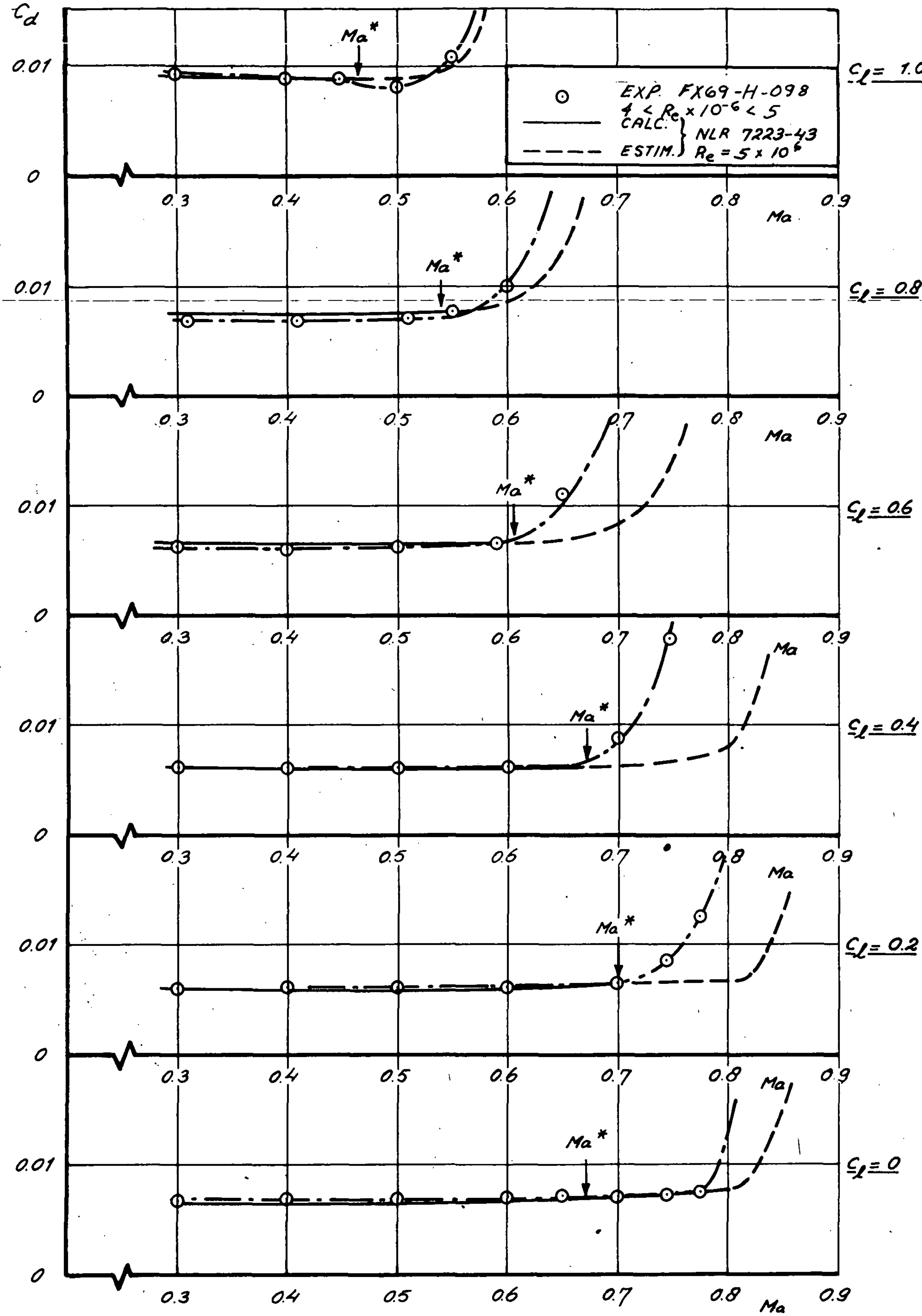


FIG. 40 AIRFOIL 2.
ESTIMATED DRAG AT CONSTANT LIFT AS A FUNCTION OF MACH NUMBER.

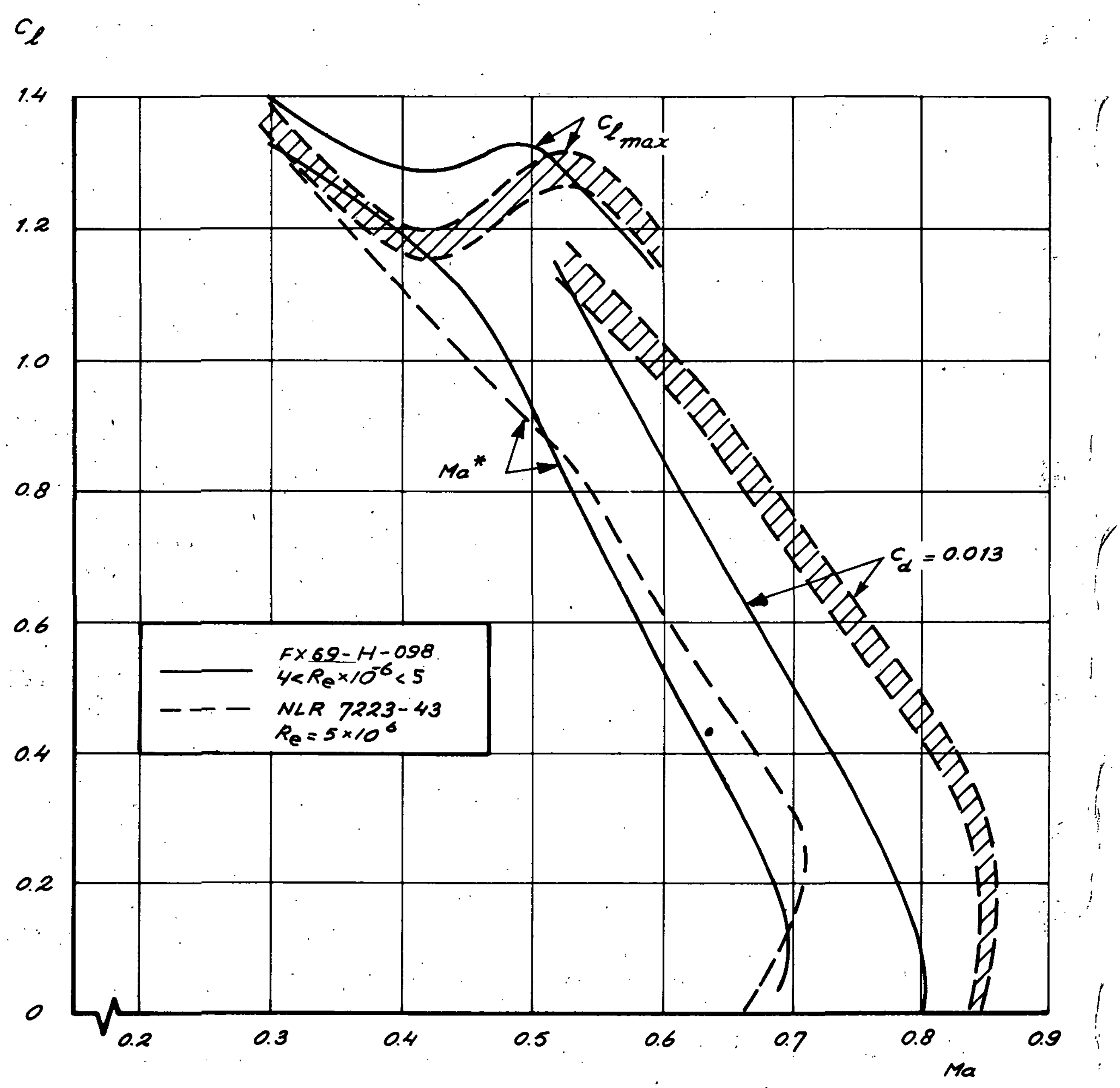


FIG. 41 AIRFOIL 2.
ESTIMATED BOUNDARIES IN c_l-Ma PLANE.

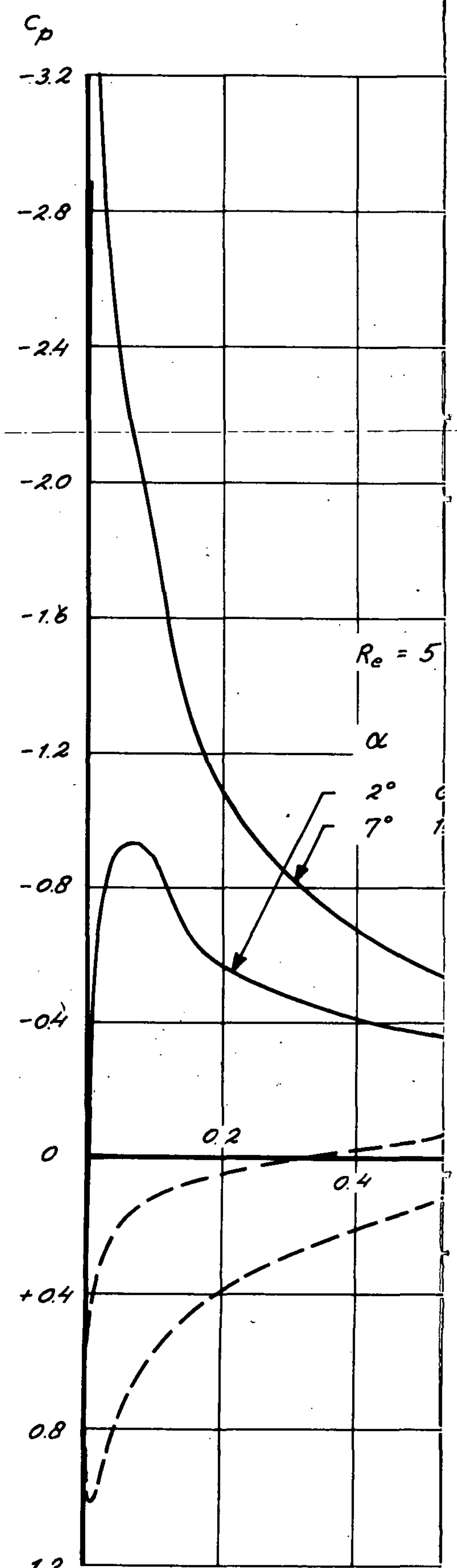


FIG. 44 AIRFOIL 2.
PRESSURE DISTRI
INCLUDING EFFEC
LAYER (APPROX. N

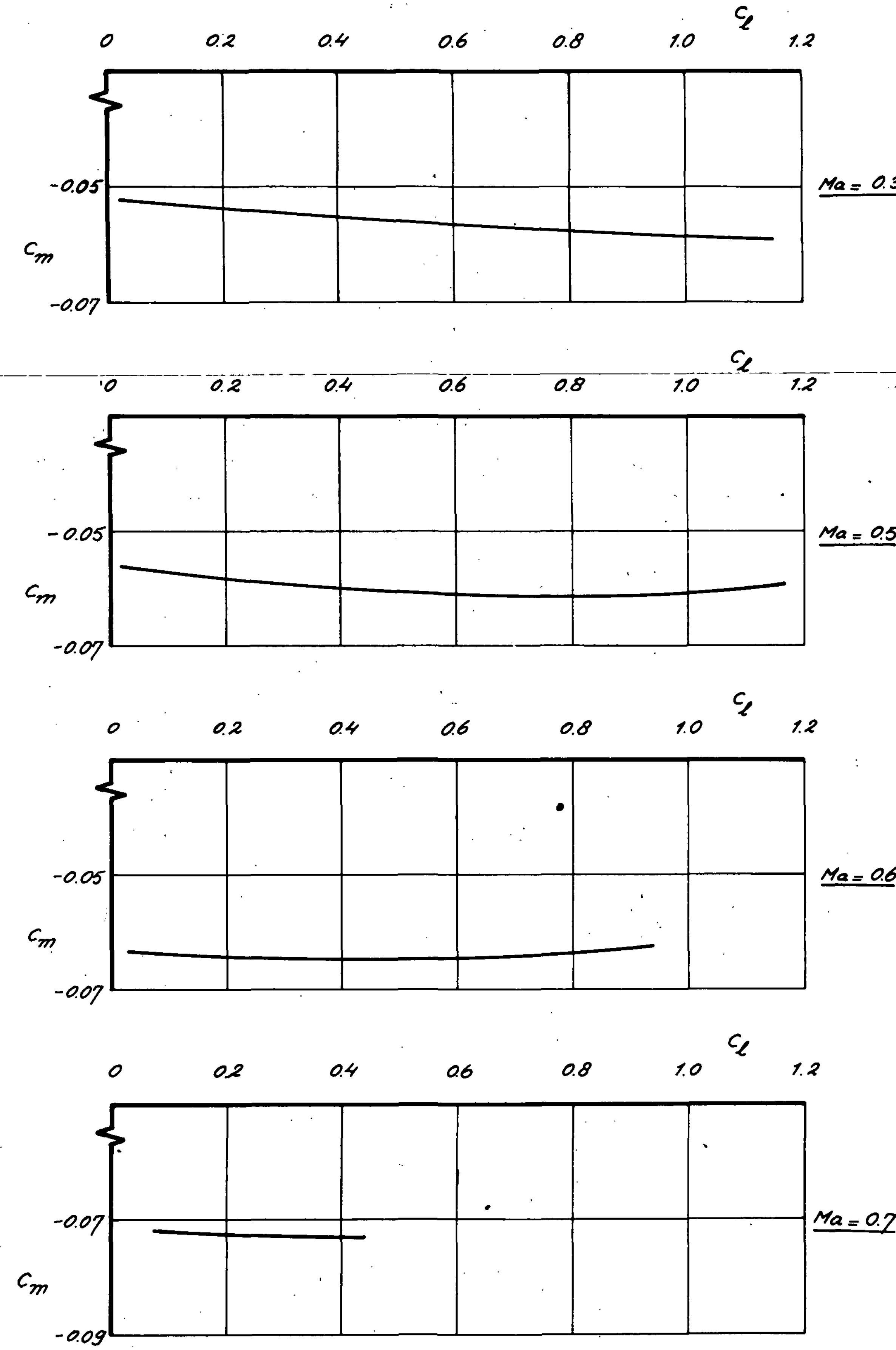


FIG. 42 AIRFOIL 2.
CALCULATED PITCHING MOMENT AS A FUNCTION OF c_ℓ FOR SEVERAL MACH NUMBERS (POTENTIAL FLOW).

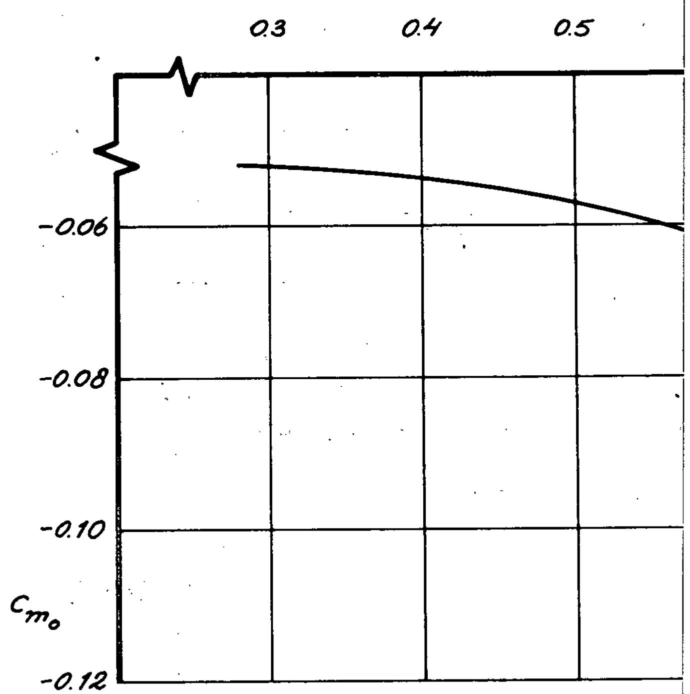


FIG. 43 AIRFOIL 2.
ZERO LIFT PITCH
A FUNCTION OF
(INVISCID FLOW)

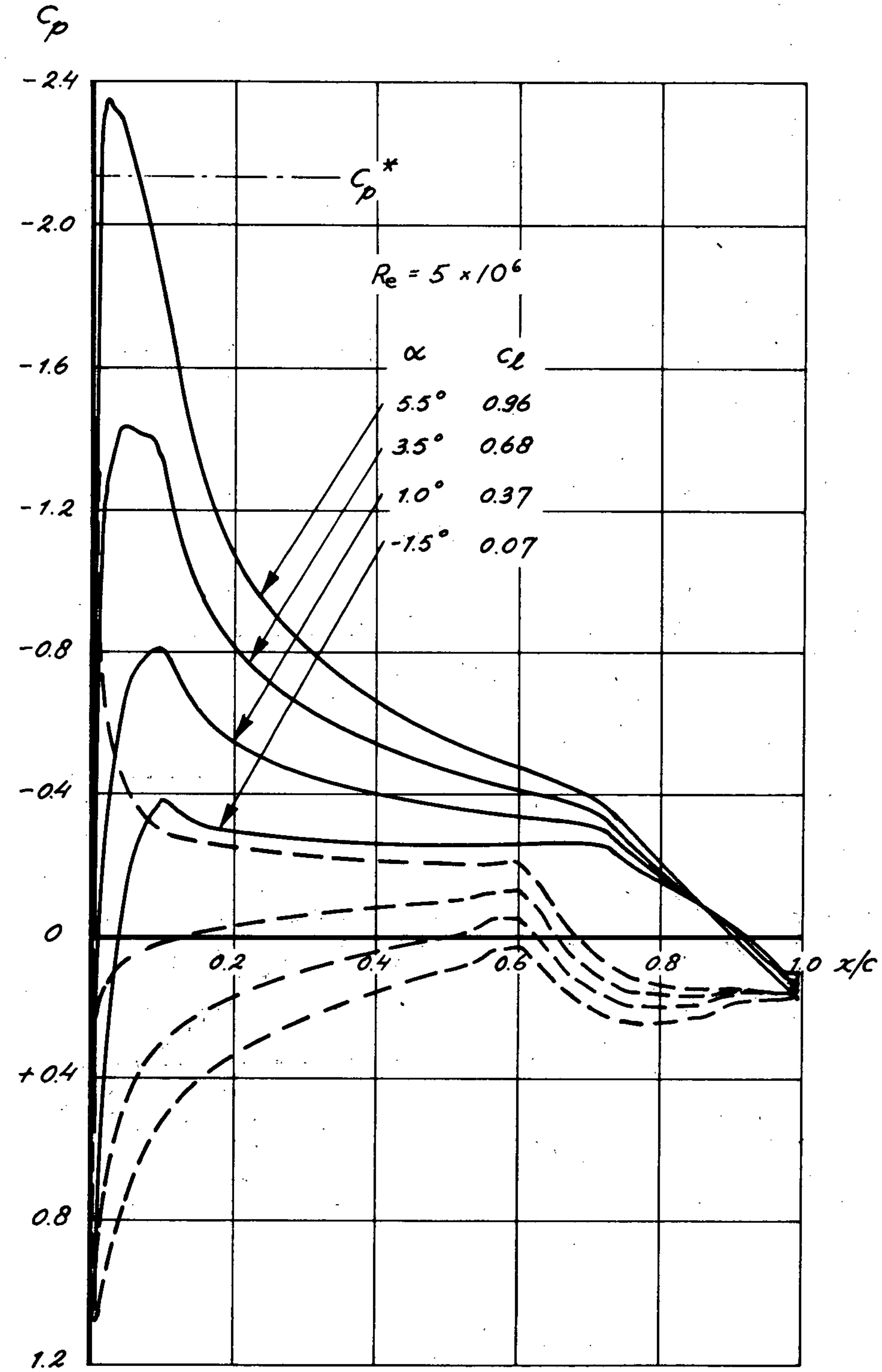


FIG. 45 AIRFOIL 2.
PRESSURE DISTRIBUTIONS AT $Ma = 0.5$,
INCLUDING EFFECT OF THE BOUNDARY
LAYER (APPROX. METHOD).

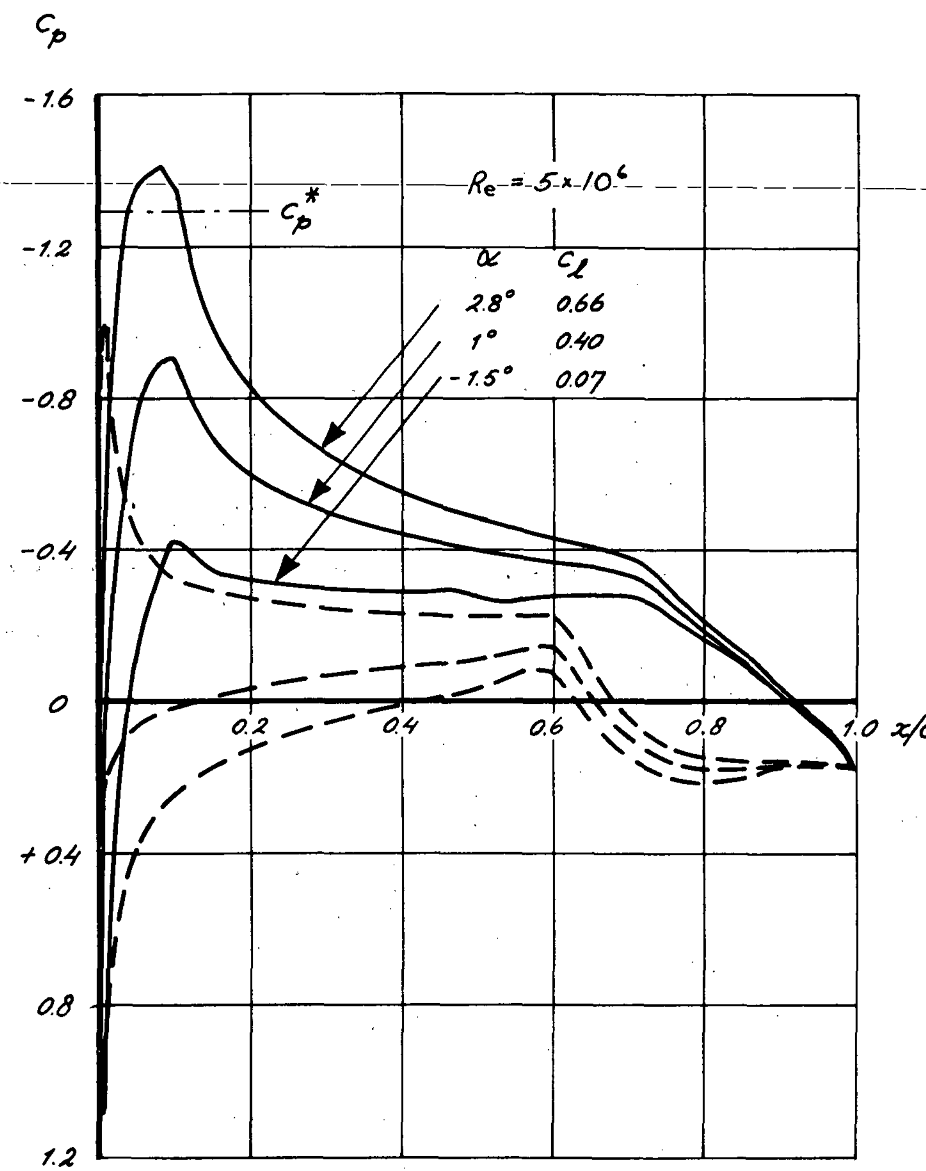


FIG. 46 AIRFOIL 2:
PRESSURE DISTRIBUTIONS AT $Ma = 0.6$, INCLUDING EFFECT OF THE BOUNDARY LAYER (APPROX. METHOD).

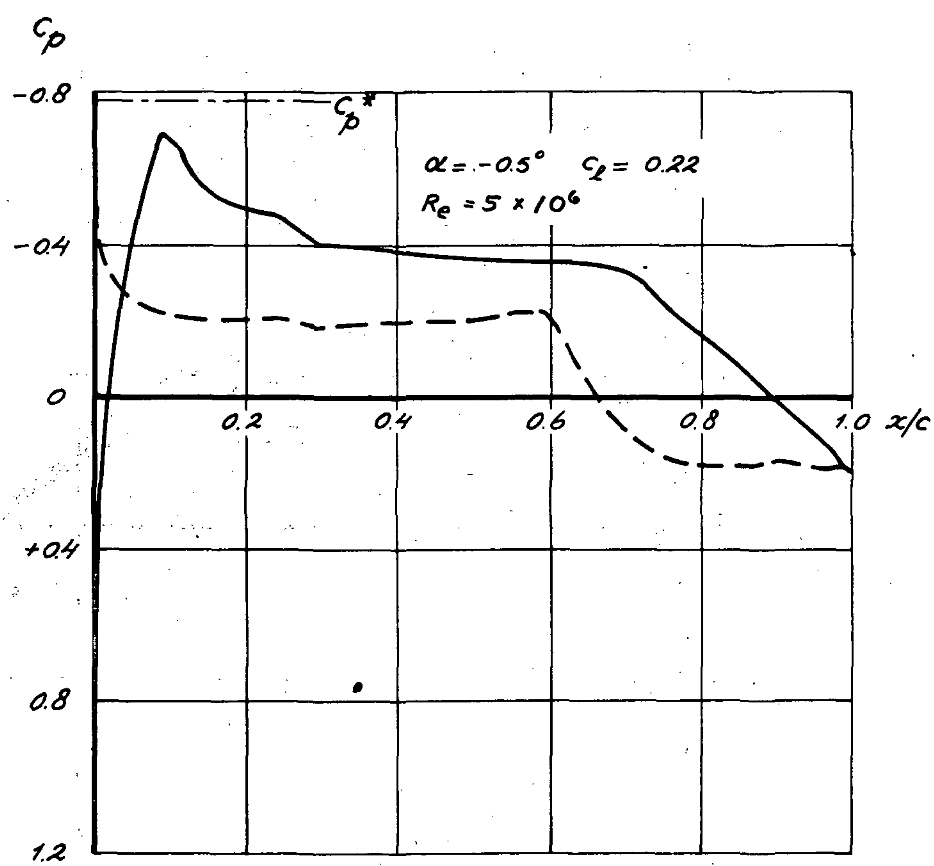


FIG. 47 AIRFOIL 2.
PRESSURE DISTRIBUTION AT $Ma = 0.7$, INCLUDING EFFECT OF THE BOUNDARY LAYER (APPROX. METHOD).

APPENDIX B

NATIONAAL LUCHT- EN RUIMTEVAARTLABORATORIUM

NATIONAL AEROSPACE LABORATORY NLR

THE NETHERLANDS

NLR TR 72128 C

ALGOL PROGRAMS FOR THE COMPUTATION OF QUASI-ELLIPTICAL SHOCK-FREE TRANSONIC AEROFOILS

by

J.W. Boerstoel and G.H. Huizing

SUMMARY

A user-oriented description of a set of ALGOL programs for the design of quasi-elliptical shock-free transonic aerofoils is presented.

CONTENTS

TR 72 128 C

LIST OF SYMBOLS

c	chord length of aerofoil
C_L	lift coefficient
$e\,[\hat{g}]$	error measure, eq. (5)
g	uncorrected interpolating function, section 2.7
$\hat{g}$	corrected interpolating function
M	local Mach number
M_∞	free stream Mach number
M_c	local Mach number at t.e.
N	highest point number, section 2.7
R	radius of curvature
$s\,[\hat{g}]$	smoothness measure, eq. (6)
t	thickness of aerofoil
$\tilde{x}$	co-ordinate
$\tilde{x}_b$	co-ordinate belonging to $\tilde{\psi}_b$
$\tilde{y}$	co-ordinate
$\tilde{y}_b$	co-ordinate belonging to $\tilde{\psi}_b$
α	incidence of ellipse in incompressible flow, section 2.2
γ	ratio of specific heats, 1.4
Γ	flow circulation
ε	weight, table 11, or: $\varepsilon_o e^{2i\alpha}$
ε_o	parameter defining thickness ratio of ellipse in incompressible flow, section 2.2
ρ_i^{-2}	denominators of eq. (6)
ζ	weight, eq. (7)
ζ_1	complex variable, see eq. 3.9 of Ref.
ζ_2	complex variable, see eq. 3.9 of Ref.
τ	velocity parameter depending only upon M, eq. (3)
τ_∞	free stream value of τ, eq. (1)
τ_c	value of τ at t.e., eq. (2)
τ_{ζ_1}	value of τ at branch point of hodograph surface
$\mu = \mu_2$	parameter in $\tilde{\psi}_c$, section 2.2
μ_{min}	value of μ when $\tilde{\psi}_c$ is made small
λ_1	nose bluntness parameter
λ_2	camber parameter
$\tilde{\psi}$	stream function

$\tilde{\Psi}_b$ stream function of basic flow

$\tilde{\Psi}_c$ correction stream function

θ flow angle

θ_c flow angle at (cusped) t.e.

θ_{ζ_1} flow angle at branch point of hodograph surface

Sub- and superscripts

$'$ first derivative

$''$ second derivative

III third derivative

IV fourth derivative

V fifth derivative

$\hat{}$ corrected data or functions, section 2.7

i point number, section 2.7

1 INTRODUCTION

Transonic shock-free lifting profiles can be developed with hodograph theory. This theory has been used in the reference to develop a calculation method for so-called quasi-elliptical aerofoils. The calculation method has been programmed. This report describes how the computer programmes have to be used in order to obtain shock-free transonic aerofoils.

The program package description is user-oriented. Information is given about some theoretical background needed (section 2), the aerofoil design process (section 3) and the computer programs (section 4). The program listings are given in the appendices.

2 THEORETICAL BACKGROUND

2.1 Scope

In order to run the program package some theoretical knowledge is needed with respect to a number of subjects. Before proceeding to a description of the handling of the programs we will first present this background information.

2.2 The parameters and the section shapes

In principle, compressible flows around q.e. aerofoils are obtained by applying a mathematical transformation to the stream function of an incompressible flow around an ellipse. The resulting stream function $\tilde{\psi}_b$ defines aerofoil section contours depending on six parameters: M_∞ (= free stream Mach number), ε_o ($2\,\varepsilon_o^{1/2}$ is the excentricity of the ellipse), α (=incidence of ellipse), Γ (= flow circulation) and the parameters λ_1 and λ_2 which control the nose shape (Fig. 1).

The section contours defined by $\tilde{\psi}_b$ have a gap at the trailing edge. In many cases this gap is so small that for engineering purposes it can be neglected.

If desired the gap can be closed by adding to $\tilde{\psi}_b$ a correction stream function $\tilde{\psi}_c$ depending upon M_∞ and three other parameters. The resulting closed aerofoil has a cusped trailing edge. The three other parameters are M_c (= local Mach number at t.e.),

θ_c (= local flow angle at t.e.) and μ, a parameter controlling the section slope in the stagnation point (see Fig. 1). When the gap is closed, λ_1 and λ_2 are not avialable as parameters (λ_1 and λ_2 have to be set zero).

The closed aerofoil sections depend thus upon seven parameters. If the closure is not desired, six parameters define the section shape.

In the computer programs M_∞ and M_c are replaced by the equivalent velocity parameters τ_∞ and τ_c, defined by

$$M_\infty^2 = \frac{2\tau_\infty}{(\gamma-1)(1-\tau_\infty)} \qquad (\gamma=1.4) \qquad (1)$$

$$M_c^2 = \frac{2\tau_c}{(\gamma-1)(1-\tau_c)} \qquad (\gamma=1.4) \qquad (2)$$

See also the conversion table 12.

A table of parameter values that may be used as a guide in selecting suitable parameter values is given in table 1.

2.3 The independent hodograph variables and the dependent physical variables.

The qualitative relation between the independent hodograph variables and the dependent physical variables is first presented for closed aerofoils.

The independent hodograph variables used are the velocity parameter τ and the flow angle θ. τ is related to Mach number by (see also table 13)

$$M^2 = \frac{2\tau}{(\gamma-1)(1-\tau)} \qquad (\gamma=1.4) \qquad (3)$$

The part of the hodograph surface that is of interest and that corresponds to a flow around a closed q.e. aerofoil consists of two sheets. The sheets are generated by a branch point $(\tau_{\zeta_1}, \theta_{\zeta_1})$ located near $(\tau_\infty, 0)$. In order to be able to distinguish between the two sheets we may introduce a cut from the branch point as indicated in Fig. 2 with the convention that a passage of the cut implies a passage to the other sheet.

TR 72/128 C

The qualitative relation between section contour and free stream lines extending in the physical $(\tilde{x},\tilde{y})$ plane to infinity from the stagnation point and from the t.e. is then as sketched in Fig. 2. On these lines the stream function $\tilde{\psi}_b + \tilde{\psi}_c = 0$. The following points are noted:

a point 1 is the l.e. stagnation point, point 6 the t.e. point, where $(\tau,\theta) = (\tau_c,\theta_c)$

b (1,2,3) is the upper front part of the section contour;
(3,4,5,6) is the upper rear part of the section contour;
(6,7,8,9) is the lower rear part of the section contour;
(9,10,1) is the lower front part of the section contour;

c the stream function $\tilde{\psi}_b + \tilde{\psi}_c$ has a singularity at $(\tau_\infty,0)$ in one of the sheets; this singularity corresponds to free stream conditions. The stream function is regular everywhere else on the hodograph surface;

d (11,1,12,13) is the free streamline from the l.e. stagnation point, (6,61,62) is the free streamline from the t.e. point.

When the aerofoil is not made closed by adding $\tilde{\psi}_c$ to $\tilde{\psi}_b$ the qualitative relation between section contour and free streamlines is as sketched in Fig. 3. The difference occurs at the saddle point of the stream function $\tilde{\psi}_b$ on the hodograph surface near the expected t.e. point image: $\tilde{\psi}_b$ is not zero in this point. In the physical surface the section contour is then not closed. The mismatch at the expected location of the t.e. is often negligible from an engineering point of view.

2.4 The closure correction and the choice of the parameters τ_c,θ_c and μ

An uncorrected aerofoil can be closed by adding to $\tilde{\psi}_b$ the correction stream function $\tilde{\psi}_c$ having as parameters τ_c,θ_c and μ. The correction is based on forcing a saddle point of $\tilde{\psi}_b + \tilde{\psi}_c$ at (τ_c,θ_c) in the second of the (τ,θ) surface (Fig. 2) on the image $\tilde{\psi}_b + \tilde{\psi}_c = 0$ of the closed aerofoil.

The parameter μ can either be specified, or if unspecified, is determined in such a way that $\tilde{\psi}_c$ is small in a certain mathematical sense (for details see the Ref.).

Experience has shown that an uncorrected aerofoil can only be closed, if the value of $\tilde{\psi}_b$ at the saddle point of $\tilde{\psi}_b$ is small

enough ($/\tilde{\psi}_b/ < 0.02$ appr.), Moreover, (τ_c, θ_c) should be given values (approximately) equal to the values of (τ, θ) in this saddle point.

A special program is available to locate the saddle point of $\tilde{\psi}_b$ if desired, so that the value of (τ, θ) and $\tilde{\psi}_b$ at the saddle point can be determined.

2.5 Limitations on the choice of the parameters

The values of the parameters determining the section shapes (see section 2.2) cannot be chosen arbitrarily. The limitations are of two different natures.

<u>a</u> Although $\tilde{\psi}_b$ and $\tilde{\psi}_b + \tilde{\psi}_c$ are one-valued functions on the two sheets of the hodograph manifold, computed results may become unacceptable when the mapping of the hodograph manifold to the $(\tilde{x}, \tilde{y})$ surface is not regular. The mapping can become singular by the appearance of <u>limit lines</u> (= folds in the supersonic parts of the physical $(\tilde{x}, \tilde{y})$ surface) or of branch points in the subsonic parts of the flow outside the aerofoil contour of the $(\tilde{x}, \tilde{y})$ surface. The singularities can only be discovered by computing the section explicitely and inspecting the results.

<u>b</u> When a <u>closed section</u> is aimed at, α and Γ should closely satisfy the relation

$$\alpha = \arcsin \frac{\Gamma}{4\pi} \qquad (4)$$

in order to obtain values of $\tilde{\psi}_b$ at the saddle point of $\tilde{\psi}_b$ near the expected t.e. position that are small enough (see section 2.4). The degree of freedom when deviating from this relation can only be established by computations and inspection of results. The rule implies that closed aerofoils are only possible for aerofoil with negligible camber and hence without rear-loading. In this case the parameters λ_1 and λ_2 have to be set equal to zero; they are thus not available for control of the section shape.

2.6 The accuracy of the computed results

The accuracy aimed at in the computations is of the order 10^{-4} for $\tilde{\psi}$, $\tilde{\psi}_\tau$, $\tilde{\psi}_\theta$, $\tilde{x}$ and $\tilde{y}$, where $\tilde{\psi}$ stands for either $\tilde{\psi}_b$ or $\tilde{\psi}_b + \tilde{\psi}_c$.

TR 72 128 C

This accuracy is in many cases not obtained when $\mathcal{T}$ is about 0.9 à 1.2 * $\mathcal{T}_\infty$ and $/\theta/ < 15^\circ$, that is on the upper surface on the rear half of the sections, and over the last 70 % of the lower surface. This is due to the use of a convergence accelerator (the ε-algorithm) applied to the series expansion for $\tilde{\psi}, \tilde{\psi}_\mathcal{T}, \tilde{\psi}_\theta, \tilde{x}$ and $\tilde{y}$. Near the singularities $(\mathcal{T}_\infty, 0)$ and $(\mathcal{T}_{\zeta_1}, \theta_{\zeta_1})$ of these series the ε-algorithm introduces errors of stochastical nature. The errors in the computed values are uncorrellated, except when in a contour point the computed value of ψ differs appreciably from zero; correlation in errors occurs in about 2 % of all data of a complete section contour.

In general the computed data are not accurate enough for engineering purposes. A smoothing correction method has to be applied to convert the data to data that are accurate enough (section 2.7).

2.7 Smoothing correction of aerofoil data

The smoothing correction of the profile data needed for reasons explained in section 2.6 is effectuated by a special method in which an error measure is carefully balanced against a smoothness measure.

The details of the smoothing correction method are as follows. Let (x_i, y_i, y_i', y_i''), $i = 0(1)N$, be given estimates of unknown points $(x_i, \vartheta_i, \vartheta_i', \vartheta_i'')$ where primes indicate first derivatives and double primes second derivatives.

Let $(\Delta y_i, \Delta y_i', \Delta y_i'')$ be given estimates of the accuracies of (y_i, y_i', y_i''). An error measure taking into account scale differences in accuracies is defined as

$$e[\mathfrak{z}] = \sum_{i=0}^{N} \left[\left(\frac{y_i - \vartheta_i}{\Delta y_i} \right)^2 + \left(\frac{y_i' - y_i'}{\Delta y_i'} \right)^2 + \left(\frac{y_i'' - \vartheta_i''}{\Delta y_i''} \right)^2 \right] \quad (5)$$

where $\mathfrak{z}$ is a function interpolating the unknown $(x_i, \vartheta_i, \vartheta_i', \vartheta_i'')$ and defined below. A smoothness increase for $\mathfrak{z}$ taking into account estimated scale differences in smoothness of $\mathfrak{z}$ is defined as

$$s[\hat{g}] = \sum_{i=1}^{N} \frac{\int_{x_{i-1}}^{x_i} \{\hat{g}''(x)\}^2 dx}{\int_{x_{i-1}}^{x_i} \{g''(x)\}^2 dx} \qquad (6)$$

where g is a function interpolating the given (x_i, y_i, y_i', y_i''). g and $\hat{g}$ are defined in each interval $[x_{i-1}, x_i]$ as a fifth degree polynomial in x interpolating $(x_{i-1}, y_{i-1}, y_{i-1}', y_{i-1}'')$ and (x_i, y_i, y_i', y_i''), respectively $(x_{i-1}, \hat{y}_{i-1}, \hat{y}_{i-1}', \hat{y}_{i-1}'')$ and $(x_i, \hat{y}_i, \hat{y}_i', \hat{y}_i'')$.

The determination of the unknown corrected points $(x_i, \hat{y}_i, \hat{y}_i', \hat{y}_i'')$ is based on the minimisation of the expression

$$\xi\, e\, [\hat{g}] + (1-\xi) s\, [\hat{g}] \qquad 0 < \xi < 1 \qquad (7)$$

where ξ is a weighting parameter which is used to balance the error and smoothness measures in such a way that $e\,[\hat{g}]$ takes approximately its expected value $3(N+1)$. (Note that the terms in $e[\hat{g}]$ should be of order 1 in the final result).

For fixed ξ the expression to be minimized is a quadratic form in the unknowns; its minimum is determined by standard methods. ξ is determined iteratively.

From the output of the aerofoil programs tables of points $(x_i, y_i, \theta_i, {}^1/R_i)$ on the section contours may be composed. These tables are converted to (x_i, y_i, y_i', y_i'') tables and corrected separately for the upper and lower half of the section contours. The accuracies $(\Delta y_i, \Delta y_i', \Delta y_i'')$ are determined by rules given in table 11 of section 4.7.2 (card input specification of the correction program).

In the smoothing correction program the correction is (slightly) biased in such a way that y_i values are more likely

corrected than y' and y_i'' values.

Vast experience with about twenty aerofoils in various situations (internal coherence of corrected data, model making and windtunnel testing, control computations with panel methods for subcritical flows) has shown that the smoothing correction method gives corrected data that are sufficiently accurate for engineering purposes, provided a redundancy of data (at least 70 points per section side) is corrected.

3 AEROFOIL DESIGN PROCESS

When a transonic shock-free q.e. aerofoil is developed, a design process has to be followed in order to fix the parameters that determine the section shape.

Experience has learned that a random approach in this design process is undesirable . In order to save efforts a certain policy has to be followed. The rules of this policy have been incorporated in the flow chart of the design process of figure 4.

Roughly speaking the design process comprises three stages. In the first stage the parameters τ_∞, ε_0, α and Γ are determined from desired values for M_∞, t/c and C_L and a desired type of loading, and a decision is made whether or not the gap at the t.e. will be closed. In the second stage the nose shape is optimized to approach desired characteristics as close as possible; this fixes the remaining parameters. In the third stage detailed computations of the section shape are performed.

During the first two stages if is necessary to check repeatedly whether limit lines and/or branch points disturb the section shape.

During a design process for an aerofoil the flow chart of the design process has to be used together with the flow chart for the data flow through the programs of Fig. 5 (section 4.9). This flow chart shows how the parameters are put into the various programs.

4 COMPUTER PROGRAMS AND NUMERICAL EXAMPLES

4.1 General description of programs

The design of a transonic shock-free q.e. aerofoil is performed with the following five ALGOL [1] CDC-6600 computer programs:

program name	program main function
CØEFF	computes tables of coefficients
INTCØNS	computes constants in compressions for $\tilde{x}_b$ and $\tilde{y}_b$
SADDPNT	searches for the saddle point of $\tilde{\Psi}_b$ near expected t.e. position on (τ,θ) surface
AIRFØIL	computes parts of aerofoil section contour and/or position of sonic lines
SMØØTH	corrects aerofoil data by special smoothing correction method

Details about the functions, inputs and outputs of these programs are given in sections 4.3 to 4.7.

Various tables of data are written to or read from a magnetic tape by the first four programs. The organisation of the data on this tape as far as needed by a user who wants to put the programs into operation is explained in section 4.8.

The listings of the programs are presented in appendix A to E.

A flow diagram presenting the flow of data through the programs is discussed in section 4.9.

The card inputs are free formatted except where specified otherwise.

1) An ALGOL compiler successfully used is the ALGOL-60 PSR302+3IC compiler of the CDC Computing Centre, Rijswijk, The Netherlands.

TR 72128 C

4.2 Numerical examples

The input and output descriptions in sections 4.3 to 4.7 are illustrated by numerical examples showing details of the input and output.

The examples should be completely and exactly reproduced by a new user of the programs in order to test all functions of the programs and/or to gain enough initial experience with the use of the programs.

4.3 Program CØEFF

4.3.1 Function

CØEFF computes the complex coefficients that are needed in INTCØNS, SADDPNT and AIRFØIL. The coefficients depend upon ε_0, α and Γ.

4.3.2 Input

The data needed by CØEFF are taken from cards. The card input specification is given in table 2. The input of the example is given in table 13.

4.3.3 Output

The results of CØEFF are output to the line printer and to the tape.

<u>a</u> The data output to the lineprinter are:

I CASE, ε_0, α, Γ.

II the complex qualities ζ_1 and ζ_2 defined in eq. (3.9) of the Ref. and the quantities $/\zeta_1/$, $/\zeta_2/$, ζ_1/ζ_2 and ε.

> N.B. The value of ζ_1 is needed for the card-input specification ror AIRFØIL, see table 7 of section 4.6.2.

III TØL, M.

IV ten integer numbers (numbers of terms in power series). The computation times of CØEFF depend approximately linearly upon the largest of these ten numbers.

V tables of the complex coefficients.

The output of the example is given in Fig. 6.

b The data output to the tape are:

I one record with M, TØL, CASE, ε_o, α, Γ.

II one record with all complex coefficients.

4.4 Program INTCØNS

4.4.1 Function

INTCØNS computes constant terms in the expressions for $\tilde{x}_b$ and $\tilde{y}_b$. The constants terms depend upon $\tau_\infty, \varepsilon_o, \alpha$ and Γ. The constants are needed in AIRFØIL.

4.4.2 Input

The data needed by INTCØNS are taken from cards and from the tape. The card input specification is given in table 3. Reading from tape is possible if CØEFF has first been used. The input of the example is given in table 13.

4.4.3 Output

The results of INTCØNS are output to the line printer. These are:

I CASE, $\varepsilon_o, \alpha, \Gamma$.

II the complex quantities ζ_1 and ζ_2 defined in eq. (3.9) of the ref. and the quantities $/\zeta_1/$, $/\zeta_2/$, ζ_1/ζ_2 and ε.

III two integer numbers (numbers of terms in power series). The computation times of INTCØNS depend approximately linearly upon the largest of these two numbers.

IV for each value of τ_∞ specified in the input:

- the value of τ_∞
- six lines with four real numbers

The 24 real numbers are the constant terms. They are needed for the card input of AIRFØIL, see section 4.6.2.

The output of the example is given in Fig. 7.

4.5 Program SADDPNT

4.5.1 Function

SADDPNT searches for the location of the minimum of the saddle point of $\tilde{\Psi}_b$ near the expected t.e. position on the (τ, θ) surface (see section 2.4). The saddle point is characterized by the relations $\tilde{\Psi}_{b_\tau} = 0$, $\tilde{\Psi}_{b_\theta} = 0$.

TR 72128C

4.5.2 Input

The data needed by SADDPNT are taken from cards and from the tape. The card input specification is given in table 4. Reading from tape is possible after the use of CØEFF. The card input of the example is given in table 13.

4.5.3 Output

The results of SADDPNT are output to the line printer. These are:

I CASE, ε_o, α, Γ, τ_∞

II at most six blocks of five lines; each line contains values of τ, θ(degrees), $\tilde{\psi}_b$, $\tilde{\psi}_{b_\tau}$ and $\tilde{\psi}_{b_\theta}$ respectively.

One block contains the results of one step in the iterative search process for the location of the saddle point.
The line after the last block contains the desired values of τ, θ, $\tilde{\psi}_b$, $\tilde{\psi}_{b_\tau}$, $\tilde{\psi}_{b_\theta}$ in the saddle point provided $/\tilde{\psi}_{b_\tau}/$ and $/\tilde{\psi}_{b_\theta}/$ are small enough (typically $<10^{-4}$, 10^{-5} respectively).
If $/\tilde{\psi}_{b_\tau}/$ and $/\tilde{\psi}_{b_\theta}/$ are not small enough a new search has to be performed starting from improved initial estimates $\tau_c^{(1)}$, $\theta_c^{(1)}$. If $\tilde{\psi}_b$ is not small in the saddle point ($/\tilde{\psi}_b/<$ about 0.02) the aerofoil cannot be closed at the t.e.

The output of the example is given in Fig. 8; $\tilde{\psi}_b$ in the saddle point is probably small enough for a successful attempt to close the gap at the t.e.

4.6 Program AIRFØIL

4.6.1 Functions

AIRFØIL computes parts of (either uncorrected or closed) aerofoil contours and /or the corresponding sonic lines.

4.6.2 Input

The data needed by AIRFØIL are taken from cards and from tape.

The pile of cards to be read consists of two parts. The first part is completely independent of the functions desired from the program; this part is specified in table 7. The second part specifies what functions of the program are desired and what calculations are to be performed. The input specification for the

second part is given in table 9.

Reading from tape is possible after the use of CØEFF. The input of the example is given in table 13.

4.6.3 Output

The results of AIRFØIL are output to the line printer. These are

I $\varepsilon_o, \alpha, T, \tau_\infty, \tau_{S_1}$, CASE, $M_\infty, \lambda_1, \lambda_2$

II if a closed aerofoil is aimed at: τ_c, θ_c, and some other quantities related to the correction stream function $\tilde{\psi}_c$.

III 24 real constants. When a closed aerofoil is not aimed at these are equal to the 24 real constants input from results of INTCØNS, see table 7.

IV data for the stagnation point including μ.

V either data for contour points or data for sonic lines.

The output of the example is given in Fig. 9 (the sonic line is situated inside the aerofoil).

4.7 Program SMØØTH

4.7.1 Function

SMØØTH corrects the aerofoil co-ordinate values obtained with AIRFØIL for large errors of stochastical nature (section 2.6) by the smoothing correction method outlined in section 2.7.

4.7.2 Input

The data required by SMØØTH are taken from cards. The card input specification and the input of the example are given in table 10.

4.7.3 Output

The results of SMØØTH are output to the line printer. They are:

I a table of the first part of the card input.

II a table of the aerofoil co-ordinate values $\{x_i, y_i, \theta_i, (^1/R)i\}$ modified to $\{x_i, y_i, y_i', y_i''\}$ values.

III a table of the second part of the card input.

IV a table of weights ρ_i^2, where the ρ_i^{-2} are equal to the denominators in the expression (6) for s $[\hat{g}]$ in section 2.7.

V for each iteration step in the iteration process on $\varepsilon = \frac{\xi}{1-\xi}$ (see section 2.7 for the meaning of ξ) information which can be used to check the course of the iteration process; this information is of interest for the analysis of details of the computation process only.

VI a table of corrected values $x_i, \hat{y}_i, \hat{y}_i', \hat{y}_i''$ together with the corrections $\hat{y}_i - y_i$, $\hat{y}_i' - y_i'$, $\hat{y}_i'' - y_i''$.

VII a table of interpolated results; in each interval $[x_i, x_{i+1}]$, $i = 0(1)N-1$, five values of $\hat{g}, \hat{g}^{I}, \hat{g}^{II}, \hat{g}^{III}, \hat{g}^{IV}$ and $\hat{g}^{V}$ are printed in order to permit an inspection of the fluctuation behaviour of the corrected interpolating curve $\hat{g}(x)$ (defined in section 2.7).

VIII a table of corrected values $x_i, \hat{y}_i, \hat{\theta}_i, (1/\hat{R})_i$ together with corrections $y_i - \hat{y}_i$, $\theta_i - \hat{\theta}_i$, $(1/R)_i - (1/\hat{R})_i$.

The results have to be inspected for correctness in two ways. The corrections $y_i - \hat{y}_i$ printed out in the last table should be of the order of magnitude of the corresponding Δy_i specified in the input and be randomly distributed in magnitude, except possibly for roughly 20 % of the values, where the corrections $y_i - \hat{y}_i$ are allowed to be two orders of magnitude larger than the corresponding accuracies. The average value of $y_i - \hat{y}_i$ is allowed to be slightly biased to non-zero values over large parts of the aerofoil contour, provided the bias is of the order of magnitude of the corresponding accuracies. Too much bias over a part of the aerofoil contour implies that the accuracies are better than assumed, and that SMØØTH has to be rerun with smaller accuracy estimates. A second way of inspecting the results is to analyze the behaviour of $\hat{g}''$ in the table of interpolated results; in each interval $[x_i, x_{i+1}]$ $\hat{g}''$ is permitted to fluctuate by an amount till about 3/4 * the average value of $\hat{g}''$ in the interval. Larger fluctuations in some interval $[x_i, x_{i+1}]$ in general indicate errors in y_i and/or y_{i+1} considerably larger than the corresponding accuracy specifications; very large fluctuation (2 or 3 * average value of $\hat{g}''$) are not permitted and should be corrected by a rerun of AIRFØIL giving improved values of the suspected y_i and/or y_{i+1} and a rerun of SMØØTH.

The output of the example is given in Fig. 10.

4.8 The magnetic tape

4.8.1 Function

The magnetic tape is used to save and to make available to the programs INTCØNS, SADDPNT and AIRFØIL tables of complex coefficients generated by the program CØEFF and tables of so-called Chaplygin functions (see the reference, appendix A for the definition of these functions).

4.8.2 Data organisation on the tape

The data on the tape are arranged in two files. The first file contains 252 records of Chaplygin functions, for each value of τ mentioned in table 12 one record. The second file contains an even number of records, which are written to the tape by the program CØEFF; each time CØEFF is used two new records are added to the second file.

The Chaplygin functions are only available for the values of τ given in table 11. This restricts in general the choice of the values of τ and τ_∞ that have to be specified in the input of the programs. Detailed rules are given in the input specification tables.

The Chaplygin functions can be made available to users by NLR upon request.

When the program package is put into operation for the first time by a user, the tape should be only provided with the first file with Chaplygin functions; after the first file two end-of-file marks must be present.

4.9 Data flow chart for the programs

The programs CØEFF, INTCØNS, SADDPNT, AIRFØIL and SMØØTH have to be used in a certain prescribed way because the latter mentioned programs use data made available by the earlier mentioned ones. A flow chart of the data showing the relation between the inputs and outputs of the programs is prescribed in Fig. 5.

TR 72 123 C

5 REFERENCE

Nieuwland, G.Y. Transonic potential flow around a family of quasi-elliptical aerofoil sections, NLR TR. T 172, 1967.

Table 1

Parameter values with corresponding main aerofoil characteristics that were fully explored.

ε_o	α	T	τ_∞	λ_1	λ_2	τ_c	θ_c	μ	M_∞	c_ℓ	t/c	aerofoil closed at t.e.?	type of press distribution
.75	.07	.95	.08	0	0	.05048	-.0579 rad.	minimal	.659	.625	11.3 %	yes	supercrit., peaky
.71	.045	.75	.095	0	0	.06070	.0504 "	"	.724	.500	11.0 %	"	" "
.71	.055	.75	.095	0	0	.06106	-.0429 "	"	.724	.502	11.0 %	"	" "
.71	.055	.75	.085	0	0	.0519	-.0436 "	"	.680	.490	12.3 %	"	" "
.71	.055	.75	.085	0	0	.0525	-.0436 "	"	.680	.49	12 %	"	" " limit line
.71	.055	.75	.085	0	0	.0523	-.0419 "	"	.680	.491	12.1 %	"	" "
.71	.045	.75	.0775	0	0	.04580	-3.2200 degr	.74056	.648	.482	12.4 %	"	sub-crit "
.625	.055	.75	.09	0	0	.05117	-2.5094 "	.97	.703	.498	14.2 %	"	super-crit "
.625	.055	.75	.09	0	0	.05117	-2.5094 "	1.09715	.703	.498	14.2 %	"	" "
.715	.012	.15	.12	0	0	–	– "	–	.826	~.10	8.2 %	no	" "
.705	-.04	.20	.1175	0	0	–	–	–	.813	~.13	8.2 %	"	" " on both sides
.575	.0275	.375	.0983	0	0	.05529	-1.2179 "	.33	.738	.248	14.1 %	yes	"
.695	-.0533	.30	.1183	0	0	–	–	–	.819	~.20	8.3 %	no	" " " " "
.695	-.0533	.30	.1183	.2	-3.0	–	–	–	.819	~.20	8.3 %	"	" " " " "
.695	-.0533	.30	.1183	0	-3.0	–	–	–	.819	~.20	8.3 %	"	" " " " "

Table 2

Card input specification for CØEFF

name of variable	advised value	comment
M	100	highest subscript of coefficients
TØL	10^{-10}	desired relative precision of coefficients
CASE	integer	sequence number for parameter combination $\varepsilon_o, \alpha,$; do not use a sequence number previously used, as otherwise the tape will be used in an erronous way by the programs.
ε_o	$0.5<\varepsilon_o<0.9$	parameter determining thickness ratio $1-\varepsilon_o/1+\varepsilon_o$ of ellipse in incompressible flow
α		incidence of ellipse in incompressible flow (radians)
Γ	$/\Gamma/ > 0.05$	flow circulation; values of Γ with $/\Gamma/ < 0.05$ increase computation times to over 30 min. on a CDC-6600 computer and impaire the accuracy of the results.

Table 3

Card input specification for INTCØNS

name of variable	advised value	comment
CASE	table 2	sequence number of parameter combination ε_0, α, $\mathcal{T}$, see table 2.
ε_0	table 2	thickness ratio parameter, see table 2
α	table 2	incidence of ellipse in incompressible flow, see table 2.
$\mathcal{T}$	table 2	flow circulation, see table 2
N		total number of values of τ_∞ subsequently specified.
$(\tau_\infty)_1$ $(\tau_\infty)_2$ ⋮ $(\tau_\infty)_N$	select values from table 12 only	when the integration constants for fixed ε_0, α, $\mathcal{T}$ may be needed for various τ_∞, all these τ_∞ values should be specified, as the computation time in fact only depends on ε_0, α, $\mathcal{T}$ and not upon τ_∞. a select only those τ_∞ values that are specified in table 11 of section 4.8 b specify the values to eight places behind decimal point and in increasing order of magnitude

Table 4

Card input specification for SADDPNT

name of variable or array	advised value	comment
FIN	table 5	array of 62 integer numbers, specified in table 5
RIM	table 6	array of 63 integer numbers, specified in table 6
CASE	table 2	sequence number of parameter combination ε_o, α, Γ, see table 2
ε_o	table 2	thickness ratio parameter, see table 2
α	table 2	incidence of ellipse in incompressible flow, see table 2
Γ	table 2	flow circulation, see table 2
τ_∞	table 3	free stream value of τ determining M_∞, taken from table 3
$\tau_c^{(1)}$		first estimate for value of τ in saddle point; if better estimates are not available take $\tau_c^{(1)} \cong 0.6 * \tau_\infty$
$\theta_c^{(1)}$		first estimate for value of θ in saddle point if better estimates are not available take $\theta_c^{(1)} = -0.7\ \alpha$ (degrees)

```
-1225711501  +1121313101  +1125721101  +1125911401  +1125711102  -1325711201
+1125711103  +1102145104  +1102254205  +1125711106  +1125611507  +1104145108
+1104254209  +1125711110  +1125611511  -2225711501  +2121313101  +2125721101
+2125911401  +2125711102  -2325711201  +2125711103  +2102145104  +2102254205
+2125711106  +2125611507  +2104145108  +2104254209  +2125711110  +2125611511
-3225711521  +3123313121  +3125721121  +3125911421  +3125711131  -3325711221
+3225711423  -3124313523  -3125721523  -3125911223  -3125711533  +3325711123
+3225611220  -3124431120  -3125612120  -3125811420  -3125611122  +3325611520
44,41,35,31,29,25,22,20,16,14,10,7,5,1,
```

TABLE 5

SPECIFICATION OF INTEGER ARRAY FIN (NEEDED IN TABLES 4 AND 7)

```
+0,+0,+0,+0,+0,+0,-2,-2,+0,+0,+0,-2,-2,+0,+0,+0,-1,-2,-2,-2,-2,
-2,+0,+0,-2,+0,+0,+0,+2,+2,+0,-1,-2,-2,-2,-2,-2,+0,+0,-2,+0,+0,
+0,+2,+2,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
```

TABLE 6

SPECIFICATION OF INTEGER ARRAY RIM (NEEDED IN TABLE 4)

Table 7

Specification of first of two piles inputs cards for AIRFØIL

name of variable or array	advised value	comment
FIN	table 7	array of 62 integer numbers, specified in table 7.
IM	table 9	array of 756 integer numbers, specified in table 9
CASE	table 2	sequence number of parameter combination ε_o, α, Γ, see table 2
ε_o	table 2	thickness ratio parameter, see table 2
α	table 2	incidence of ellipse in incompressible flow, see table 2
Γ	table 2	flow circulation, see table 2
Re ζ_1	output of CØEFF	real part of complex constant ζ_1; this number has to be taken from the line printer results of CØEFF, see the output description in section 4.3.3, point a II; specify Re ζ_1 to all decimal places output by CØEFF
Im ζ_1	output of CØEFF	imaginary part of complex constant ζ_1; see comment after Re ζ_1.
λ_1	O(1)	real parameter λ_1 governing nose bluntness; standard value is zero
λ_2	O(1)	real parameter λ_2 governing camber; standard value is zero
T_E	1 or 0	$T_E = 1$: aerofoil will be closed at t.e. position $T_E = 0$: uncorrected aerofoil (aerofoil will have gap at expected t.e. position).
TØL	10^{-5}	precision required in various tests
MAX IT	$\geqslant 10$	maximum number of steps in various iteration processes
τ_∞	table 3	free stream value of τ determining M_∞, taken from table 3 (τ_∞ has to be taken from table 11).

Table 7 (continued)

Specification of first of two piles inputs cards for AIRFØIL

name of variable or array	advised value	comments
24 constants	output of INTCØNS	24 real constants; these constants have to be taken from the line printer results of INTCØNS, see the output description in section 4.4.3, point IV. The four numbers of each of the six lines mentioned there should be punched on one card in the order as they are printed out; the six cards obtained so have to be placed in the same order as the six lines.
τ_c	output of SADDPNT	omit this number if $T_E = 0$ (if aerofoil has gap at t.e.). τ_c is the value of τ at the t.e.; the value has to be equal to or very close to the value of τ in the saddle point of $\tilde{\psi}_b$ found with SADDPNT, see the output description of SADDPNT, section 4.5.3 point II; τ_c need not to be taken from table 11
θ_c	output of SADDPNT	omit this number if $T_E = 0$ (if aerofoil has gap at t.e.). θ_c is the value of θ at the t.e.; the value has to be equal to or very close to the value of θ in the saddle point of $\tilde{\psi}_b$ found with SADDPNT, see the output description of SADDPNT, section 4.5.3 point II.
μ		omit this number if $T_E = 0$ (if aerofoil has gap at t.e.). $\mu = 0$: $\tilde{\psi}_c$ will be made small $\mu \neq 0$: μ should approximately equal to the value of μ_{min} printed out by AIRFØIL after a run of AIRFØIL in which $\tilde{\psi}_c$ has been made small by setting $\mu = 0$; for the value of μ_{min} meant see the output description of AERFØIL section 4.6.3 point IV.

```
+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,
+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+2,-2,-2,-2,+0,+2,-2,-2,-2,+0,+0,+0,-1,-2,-2,-2,-2,
-2,+0,+0,-2,+0,+0,+0,+2,+2,+0,-1,-2,-2,-2,-2,-2,+0,+0,-2,+0,+0,
+0,+2,+2,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+2,-2,-2,-2,+0,+2,-2,-2,-2,+0,+0,+0,-1,-2,-2,-2,-2,
-2,+0,+0,-2,+0,+0,+0,+2,+2,+0,-1,-2,-2,-2,-2,-2,+0,+0,-2,+0,+0,
+0,+2,+2,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,
+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+0,+0,-1,-1,-1,+0,+0,-1,-1,-1,+0,-1,+0,-1,-1,-1,-1,
-1,+1,+1,+0,+1,+0,+0,+0,+0,-1,+0,-1,-1,-1,-1,-1,+1,+1,+0,+1,+0,
+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+2,-2,-1,-1,-1,+2,-2,-1,-1,-1,+0,-1,+0,-1,-1,-1,-1,
-1,+1,+1,-2,-1,+0,+0,+2,+2,-1,+0,-1,-1,-1,-1,-1,+1,+1,-2,-1,+0,
+0,+2,+2,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+2,-2,-1,-1,+1,+2,-2,-1,-1,+1,+0,+1,+0,-1,-1,-1,-1,
-1,-1,-1,-2,-1,+0,+0,+2,+2,+1,+0,-1,-1,-1,-1,-1,-1,-1,-2,-1,+0,
+0,+2,+2,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+0,+0,-1,-1,+1,+0,+0,-1,-1,+1,+0,+1,+0,-1,-1,-1,-1,
-1,-1,-1,+0,+1,+0,+0,+0,+0,+1,+0,-1,-1,-1,-1,-1,-1,-1,+0,+1,+0,
+0,+0,+0,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+1,+1,-1,-1,+1,-1,-1,-1,-1,+1,-1,-1,-1,-1,+1,-1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,+1,+1,+1,+1,-1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,+1,
+1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+1,+1,-1,-1,+1,-1,-1,-1,-1,+1,-1,-1,-1,-1,+1,-1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,+1,+1,+1,+1,-1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,+1,
+1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+1,+1,-1,-1,+1,-1,-1,-1,+1,+1,-1,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,-1,
-1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+1,+1,-1,-1,+1,-1,-1,-1,+1,+1,-1,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,-1,
-1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+0,+0,+1,+0,+0,+0,+0,+1,+0,+0,+1,+1
-2,-2,-1,+0,-2,-2,-2,-1,+0,+2,+1,+1
-2,-2,-1,+0,-2,-2,-2,-1,+0,+2,+1,+1
+0,+0,+1,+0,+0,+0,+0,+1,+0,+0,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
```

TABLE 8

SPECIFICATION OF INTEGER ARRAY IM (NEEDED IN TABLE 7)

Table 9

Specification of last of two piles input cards for AIRFØIL

(first part see table 7)

<table>
<tr><th>name of variable</th><th>advised value</th><th>comment</th></tr>
<tr><td>XY</td><td>UF
UR
LF
LR
US
LS
TP</td><td>two symbols to be punched in the first and second position of a card

The two symbols may take various values. They specify the functions desired from AIRFØIL

symbol value: desired function:
X = U perform computations for upper part of aerofoil
X = L perform computations for lower part of aerofoil
Y = F compute points on front part of aerofoil contour
Y = R compute points on rear part of aerofoil contour
Y = S compute points on sonic line
XY = TP compute t.e. point (only for closed aerofoils, $T_E=1$)</td></tr>
<tr><td colspan="3">if Y=F or R (computation of points on aerofoil contour)</td></tr>
<tr><td></td><td></td><td>1 For each card beginning with UF,UR,LF or LR points on an aerofoil contour will be computed. This occurs by iterative processes on θ for a sequence of increasing values of τ.

2 The sequence of values of τ is, in principle, defined by $\tau = \tau_{begin}\ (\Delta\tau)\ \tau_{end}$ where τ_{begin}, τ_{end} and the step size $\Delta\tau$ are given. However in order to obtain better density distributions of points the step size can be varied by AIRFØIL between two limits $\Delta\tau_{min}$ and $\Delta\tau_{max}$.

3 The step size in part of the iterative processes on θ is $\Delta\theta$.
4 The iteration process starts at $\tau=\tau_{begin}$, $\theta=\theta_{begin}$, where θ_{begin} is a rough estimate of the value of θ on the aerofoil contour at $\tau=\tau_{begin}$
5 The step size $\Delta\theta$ for $\tau=\tau_{begin}$ may be enlarged to $\Delta\theta_{init\ i}$ in order to permit rough guesses of θ_{begin} ($\pm$ 30° error).</td></tr>
</table>

Table 9 (Cont'ed)

name of variable	advised value	comment
		6 τ_{begin}, τ_{end} and the step sizes $\Delta\tau$, $\Delta\tau_{min}$ and $\Delta\tau_{max}$ should be for the two ranges of τ indicated below integral multiples of the values listed to the right. range of τ: step sizes multiples of: $0.0<\tau\leqslant 0.25$ $1/1200 = 0.0008\cancel{3}$ $0.25\leqslant\tau\leqslant 0.32$ $1/100 = 0.01$ This implies that for $\Delta\tau_{min} < 0.01$ it may be necessary to perform seperate calculations for each of the two τ ranges.
τ_{begin}	>0	first value of τ for which an aerofoil point has to be computed (remarks 2 and 6); specify at least 8 places behind decimal point.
$\Delta\tau$	>0	nominal step size in τ (remarks 2 and 6); specify at least 8 places behind decimal point
τ_{end}	>0	last value of τ for which an aerofoil point has to be computed (remarks 2 and 6); $\tau_{end}\geqslant\tau_{begin}$; specify at least 8 places behind decimal point.
$\Delta\tau_{min}$	>0	minimum step size in τ (remarks 2 and 6); $\Delta\tau\geqslant\Delta\tau_{min}$; specify at least 8 places behind decimal point.
$\Delta\tau_{max}$	>0	maximum step size in τ (remarks 2 and 6); $\Delta\tau\geqslant\Delta\tau_{max}$; specify at least 8 places behind decimal point.
θ_{begin} (degrees)		estimate of θ in degrees in first aerofoil point for $\tau=\tau_{begin}$ (remarks 4 and 5); the sign of θ_{begin} depends upon UF,UR,LF,LR and has to be chosen in accordance with the sign conventions for θ indicated in figures 2 and 3 of section 2.3.
$\Delta\theta_{init}$ (degrees)	3.0	initial step size for θ for $\tau=\tau_{begin}$ (remarks 4 and 5)
$\Delta\theta$ (degrees)	$\leqslant 1.0$	step size for θ for $\tau\neq\tau_{begin}$ (remark 3)
		7 If τ_{end} exceeds the maximum value of τ in a suction peak at the upper or lower side the program finishes the computations before τ_{end} at approximately this maximum value of τ.

Table 9

name of variable	advised value	comment
		8 Card examples for coarse distributions of points (assume $\tau_\infty = .10$) UF .01 .02 .32 .01 .01 +85.0 3.0 1.0 UR .08 .01 .32 .01 .01 -15.0 3.0 1.0 LF .01 .01 .25 .01 .01 -85.0 3.0 1.0 LR .08 .01 .25 .01 .01 +10.0 3.0 1.0 UR .07 .01 .07 .01 .01 -15.0 3.0 1.0(one point only) Card examples for fine distribution of points on entire lower side of aerofoil contour. LF .01 .01 .06 .01 .01 -85.0 3.0 1.0 LF .062 .01 .25 .000833333 .01 -30.0 3.0 1.0 LR .08 .005 .25 .000833333 .01 +10.0 3.0 1.0 LR .075 .01 .075 .01 .01 +10.0 3.0 1.0 LR .07 .01 .07 .01 .01 +10.0 3.0 1.0 (one point only) LR .065 .01 .065 .01 .01 +10.0 3.0 1.0 (one point only) LR .06 .01 .06 .01 .01 +10.0 3.0 1.0 (one point only)
		if Y=S (computation of sonic lines)
		1 For each card beginning with US or LS a sequence of values of θ are specified for which sonic points will be computed. The value of τ is 1/6. 2 Perform sonic line calculations after section contour calculations so that the sonic line values of θ on the section contour can be estimated. The flow field values of θ on the sonic line lie between these two values 3 A good density distribution of points is obtained if the intervals in θ are chosen as indicated by the following table. $/\theta/ > 25^\circ$: $\Delta\theta = 5^\circ$ $25^\circ > /\theta/ > 12.5^\circ$: $\Delta\theta = 2.5^\circ$ $12.5^\circ > /\theta/ > 5^\circ$: $\Delta\theta = 1.25^\circ$ $5^\circ > /\theta/ > 1.5^\circ$: $\Delta\theta = 0.5^\circ$ $1.5^\circ > /\theta/$: $\Delta\theta = 0.25^\circ$ 4 The signs of θ depend upon US,LS and have to be chosen in correspondence with the sign convention of θ indicated in figures 2 and 3 of section 2.3
N_θ	> 0	number of values of θ to be specified
θ_1		first value of θ on sonic line (degrees)
θ_2		second value of θ on sonic line (degrees)
⋮		
θ_{N_θ}		last value of θ on sonic line (degrees)

Table 9

name of variable	advised value	comment
		5 Example for upper side of airofoil for coarse density of points US 19 40.0 30.0 20.0 15.0 12.5 10.0 8.0 6.0 4.0 3.0 2.0 1.0 0.0 -1.0 -2.0 -3.0 -4.0 -6.0 -8.0
		if XY = TP (computation of t.e. point for closed aerofoils, T_E=1).
τ θ (degrees)	τ_c θ_c	choose $\tau = \tau_c$ see table 7 choose $\theta = \theta_c$, see table 7
		After having terminated the specifications for XY = UF,UR,LF,LR, or for XY=US,LS, or for XY=TP a card specifying new values for XY may be defined, or the input of cards may be terminated.

Table 10

Card input specification for SMØØTH

name of variable or array	advised value	example value	comment
i_1	$i_1 > 0$	1601	identifying sequence number for first part of card input
N	$N > 0$	33	maximum subscript i of the given points $\{x_i, y_i, \theta_i, (1/R)_i\}$ The points are numbered from 0 to N inclusive
$\{x_o, y_o, \theta_o, (1/R_o)\}$	$x_o < x_1$	SEE FIG. 10, AFTER MESSAGE DATA INPUT TAPE NUMBER 1601	$\{x_i, y_i, \theta_i, (1/R)_i\}$ are the x, y, θ (in radians) and 1/R values
$\{x_1, y_1 \theta_1, (1/R)_1\}$	$x_1 < x_2$		obtained from AIRFØIL, arranged from l.e. to t.e. The points are numbered from zero to N. The upper and lower side of an aerofoil have to be corrected by two
$\{x_N, y_N, \theta_N, (1/R)_N\}$	$x_{N-1} < x_N$		seperate runs of SMØØTH. Avoid $/\theta/ = \pi/2$. See remark 3 for t.e. derivative
i_2	$i_2 > 0$	1602	identifying integer sequence number for second part of card input
ε	$\varepsilon = 1.0$	1.0	first estimate of the weight parameter $\varepsilon = \frac{\xi}{1-\xi}$, c.f. section 2.7 for the meaning of ξ.
ϑ_o	0.01	0.01	weight, increasing accuracy of all y_i by factor $\vartheta_o^{-1/2}$
ϑ_1	0.1	0.1	weight, increasing accuracy of all y_i' by factor $\vartheta_1^{-1/2}$
ϑ_2	1.0	1.0	weight, increasing accuracy of all y_i'' by factor $\vartheta_2^{-1/2}$
μ_3	1.0	1.0	fixed real dummy numbers
μ_4	1.0	1.0	
μ_5	1.0	1.0	
N_σ		6	number of different accuracy combinations $\{(\Delta y)^2_j, (\Delta y')^2_j, (\Delta y'')^2_j, j\}$ specified subsequently below
$K_{k\ max}$	1	1	fixed integer dummy number
K_{max}	10	10	maximum number of iterations to desired value of ε permitted
tol_1	10^{-5}	10^{-5}	desired relative precision of corrected results.

tol_2	10^{-5}	10^{-5}	desired absolute precision of corrected results.
$\{(\Delta y)^2_1, (\Delta y')^2_1, (\Delta y'')^2_1, j_1\}$		SEE FIG. 10 AFTER SIGMA 0, SIGMA 1, SIGMA 2.	squared estimates of accuracies Δy, $\Delta y'$ and $\Delta y''$ of y, y', y'' $\cong$ squared estimates of accuracies of $y, \theta, 1/R$ for the points 0 to j_1 inclusive (remark 1).
$\{(\Delta y)^2_2, (\Delta y')^2_2, (\Delta y'')^2_2, j_2\}$			squared estimates of accuracies Δy, $\Delta y'$ and $\Delta y''$ of y, y', y'' for the points j_1+1 to j_2 inclusive (remark 2).
$\{(\Delta y)^2_{N_\sigma}, (\Delta y')^2_{N_\sigma}, (\Delta y'')^2_{N_\sigma}, j_{N_\sigma}\}$			squared estimates of accuracies Δy, $\Delta y'$ and $\Delta y''$ of y, y', y'' for the points $j_{N_\sigma -1}+1$ to j_{N_σ}=N (remarks 2 and 3).
			Experience has shown that the following rules for the accuracies give in general good results. Sometimes inspection of the output of SMØØTH (see section 4.7.3) may suggest better estimates of accuracies that improve the results, however. 1 The data of the points on the section nose where $\tau <$ about 0.7 0.7x τ_∞ are free of errors. By specifying the accuracy of these data equal to 10^{-15} unnecessary correction of these data is avoided. The accuracy specification of these first, say 7, points is therefore $\{10^{-30}, 10^{-30}, 10^{-30}, 6\}$ 2 The other data have in general the following accuracies:

			$/\tau-\tau_\infty/ > .04$: $\Delta y = 10^{-4}$ on upper side of aerofoil $/\tau-\tau_\infty/ \leqslant .04$: $\Delta y = 10^{-3}$ on lower side of aerofoil front part of aerofoil: $\Delta y' = 10^{-3}$ $\Delta y'' = 10^{-2}$ on front part of aerofoil $\Delta y'' = 10^{-3}$ on rear part of aerofoil $/\tau-\tau_\infty/ < .04$, rear part of aerofoil: $\Delta y = 10^{-2}$, $\Delta y' = 10^{-2}$, $\Delta y'' = 10^{-3}$
			The values of Δy, $\Delta y'$ and $\Delta y''$ are uncritical; they are allowed to be in error by factors 10. 3 In order to guarantee that upper and lower parts of the corrected aerofoil accurately match at the t.e. (these parts have to be corrected by separate runs of SMØØTH) the accuracies of the tail point should be specified as follows: $\Delta y = 10^{-15}$(no correction on y values), $\Delta y' = 10^{-3}$, $\Delta y'' = 10^{+15}$ (values of curvature in tail point are unknown; the guessed value has no accuracy). For the input value of 1/R in the tail point one may take a rough guess obtained by extrapolation. 4 A full example of the accuracy specification could be: 10^{-30} 10^{-30} 10^{-30} 6 (first seven points on nose) 10^{-8} 10^{-6} 10^{-4} 12 (accuracies of points 7 to 12 on nose) 10^{-8} 10^{-6} 10^{-6} 50 (accuracies of point 13 to 50, $/\tau-\tau_\infty/ > 0.04$) 10^{-4} 10^{-4} 10^{-6} 60 (accuracies of point 51 to 60, $/\tau-\tau_\infty/ < 0.04$) 10^{-8} 10^{-6} 10^{-6} 69 (accuracies of point 61 to 69, $/\tau-\tau_\infty/ > 0.04$) 10^{-15} 10^{-6} 10^{+15} 70 (accuracies for tail point) In this case $N_\sigma = 6$.
1.0 1.0 1.0		1.0 1.0 1.0	three fixed real numbers
-1		-1	fixed integer number
i_1	>0 <0	-1	$i_1 < 0$: termination of input specification and computations; $i_1 > 0$: continue input specification on second line of this table with N.

Table 11

Values of τ for which Chaplygin functions are available on tape.

$.01(.01).05$
$.05(^1/1200 = .0008\not{3}).1658\not{3}$
$1/6$
$.1675(^1/1200 = .0008\not{3}).25$
$.25(.01).32$

Table 12

Conversion table for τ values to M values

τ	M	τ	M
.01	0.2247	.21	1.1529
.02	0.3194	.22	1.1875
.03	0.3932	.23	1.2221
.04	0.4564	.24	1.2566
.05	0.5130	.25	1.2910
.06	0.5659	.26	1.3254
.07	0.6135	.27	1.3599
.08	0.6594	.28	1.3944
.09	0.7032	.29	1.4291
.10	0.7454	.30	1.4639
.11	0.7861	.31	1.4988
.12	0.8257	.32	1.5339
.13	0.8644		
.14	0.9022		
.15	0.9393		
.16	0.9759		
$^1/6$	1.0000		
.17	1.0120		
.18	1.0476		
.19	1.0830		
.20	1.1180		

```
CARD INPUT COEFF..

100  *-10  18  0.71  0.045  0.75


CARD INPUT INICONS..

18  0.71  0.045  0.75
7
0.075
0.075833333
0.076666667
0.0775
0.078333333
0.079166667
0.08


CARD INPUT SADDPNT..

-1225711501  +1121313101  +1125721101  +1125911401  +1125711102  -1325711201
+1125711103  +1102145104  +1102254205  +1125711106  +1125611507  +1104145108
+1104254209  +1125711110  +1125611511  -2225711501  +2121313101  +2125721101
+2125911401  +2125711102  -2325711201  +2125711103  +2102145104  +2102254205
+2125711106  +2125611507  +2104145109  +2104254209  +2125711110  +2125611511
-3225711521  +3123313121  +3125721121  +3125911421  +3125711131  -3325711221
+3225711423  -3124313523  -3125721523  -3125911223  -3125711533  +3325711123
+3225611220  -3124431120  -3125612120  -3125811420  -3125611122  +3325611520
44,41,35,31,29,25,22,20,16,14,10,7,5,1,
+0,+0,+0,+0,+0,+0,-2,-2,+0,+0,+0,-2,-2,+0,+0,+0,-1,-2,-2,-2,-2,
-2,+0,+0,-2,+0,+0,+0,+2,+2,+0,-1,-2,-2,-2,-2,-2,+0,+0,-2,+0,+0,
+0,+2,+2,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
18  0.71  0.045  0.75
0.0775  0.0465  -3.30

CARD INPUT AIRFOIL..

-1225711501  +1121313101  +1125721101  +1125911401  +1125711102  -1325711201
+1125711103  +1102145104  +1102254205  +1125711106  +1125611507  +1104145108
+1104254209  +1125711110  +1125611511  -2225711501  +2121313101  +2125721101
+2125911401  +2125711102  -2325711201  +2125711103  +2102145104  +2102254205
+2125711106  +2125611507  +2104145108  +2104254209  +2125711110  +2125611511
-3225711521  +3123313121  +3125721121  +3125911421  +3125711131  -3325711221
+3225711423  -3124313523  -3125721523  -3125911223  -3125711533  +3325711123
+3225611220  -3124431120  -3125612120  -3125811420  -3125611122  +3325611520
44,41,35,31,29,25,22,20,16,14,10,7,5,1,
+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,
+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+2,-2,-2,-2,+0,+2,-2,-2,-2,+0,+0,+0,-1,-2,-2,-2,-2,
-2,+0,+0,-2,+0,+0,+0,+2,+2,+0,-1,-2,-2,-2,-2,-2,+0,+0,-2,+0,+0,
+0,+2,+2,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+2,-2,-2,-2,+0,+2,-2,-2,-2,+0,+0,+0,-1,-2,-2,-2,-2,
-2,+0,+0,-2,+0,+0,+0,+2,+2,+0,-1,-2,-2,-2,-2,-2,+0,+0,-2,+0,+0,
+0,+2,+2,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,
+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+0,+0,-1,-1,-1,+0,+0,-1,-1,-1,+0,-1,+0,-1,-1,-1,-1,
-1,+1,+1,+0,+1,+0,+0,+0,+0,-1,+0,-1,-1,-1,-1,-1,+1,+1,+0,+1,+0,
+0,+0,+0,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+2,-2,-1,-1,-1,+2,-2,-1,-1,-1,+0,-1,+0,-1,-1,-1,-1,
-1,+1,+1,-2,-1,+0,+0,+2,+2,-1,+0,-1,-1,-1,-1,-1,+1,+1,-2,-1,+0,
+0,+2,+2,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+0,+0,+0,+0,+2,-2,-1,-1,+1,+2,-2,-1,-1,+1,+0,+1,+0,-1,-1,-1,-1,
-1,-1,-1,-2,-1,+0,+0,+2,+2,+1,+0,-1,-1,-1,-1,-1,-1,-1,-2,-1,+0,
+0,+2,+2,+0,+1,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,+0,
+0,+0,+0,-2,+0,+0,-1,-1,+1,+0,+0,-1,-1,+1,+0,+1,+0,-1,-1,-1,-1,
-1,-1,-1,+0,+1,+0,+0,+0,+0,+1,+0,-1,-1,-1,-1,-1,-1,-1,+0,+1,+0,
+0,+0,+0,-2,+1,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,+2,+0,+0,+0,+0,+0,
+1,+1,-1,-1,+1,-1,-1,-1,-1,+1,-1,-1,-1,-1,+1,-1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,+1,+1,+1,+1,-1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,+1,
+1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+1,+1,-1,-1,+1,-1,-1,-1,-1,+1,-1,-1,-1,-1,+1,-1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,+1,+1,+1,+1,-1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,+1,
+1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+1,+1,-1,-1,+1,-1,-1,-1,+1,+1,-1,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,-1,
-1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+1,+1,-1,-1,+1,-1,-1,-1,+1,+1,-1,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,
-1,+0,+0,-1,+0,-1,-1,+1,+1,+1,+0,-1,-1,-1,-1,-1,+0,+0,-1,+0,-1,
-1,+1,+1,-1,+0,-1,-1,-1,-1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,+1,
+0,+0,+1,+0,+0,+0,+0,+1,+0,+0,+1,+1
-2,-2,-1,+0,-2,-2,-2,-1,+0,+2,+1,+1
-2,-2,-1,+0,-2,-2,-2,-1,+0,+2,+1,+1
+0,+0,+1,+0,+0,+0,+0,+1,+0,+0,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,-1,-1,-1,-1,+0,+1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1
-1,-1,+0,+1,-1,-1,-1,+0,-1,+1,+1,+1


18  0.71  0.045  0.75
0.98863112240  -0.0023226491951
0.0  0.0
1  *-5  15
0.0775
+1.492885  +0.043738  +1.492885  -0.043738
-0.031692  +0.847810  +0.031692  -0.347810
-0.031692  +0.859666  +0.031692  -0.835953
-1.492885  -0.043738  +1.492885  +0.043738
-0.031692  +0.847810  +0.031692  -0.847810
-0.031692  +0.859666  +0.031692  -0.835953
0.0458  -3.22002  0.740557
UR  0.12  0.01  0.20  0.01  0.005  -12.0  2.0  1.0
TP  0.0458  -3.22002
JS  10  0.5  1.5  3  5  8  12  16  20  25  30
```

TABLE 13

CARD INPUTS OF EXAMPLES OF FIG. 6 TO 9

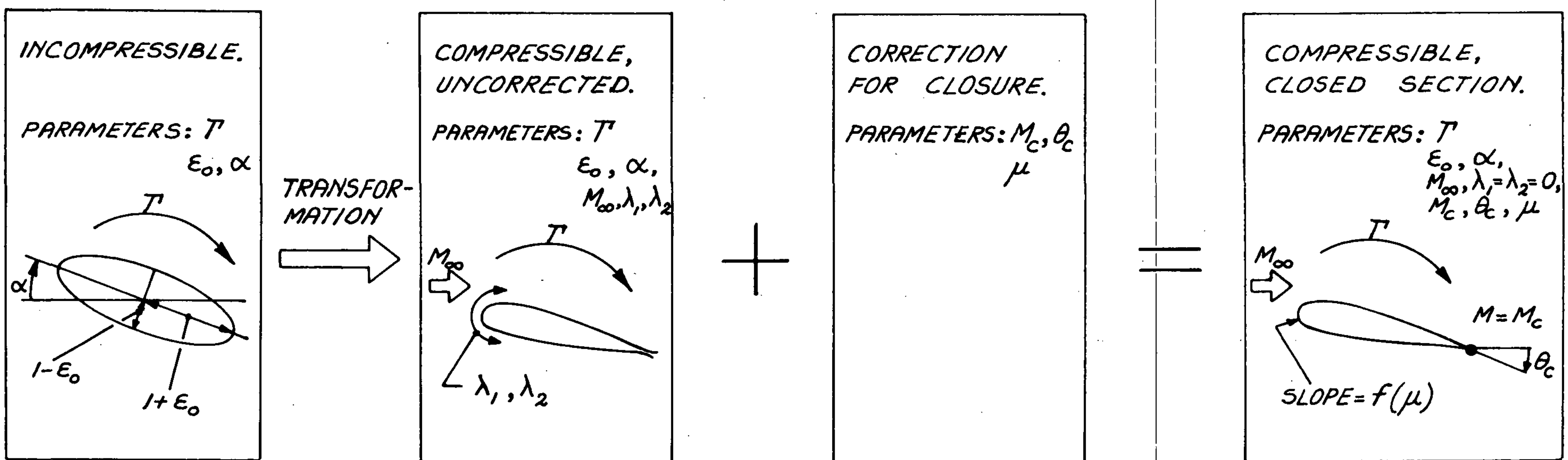


FIG. I THE PARAMETERS

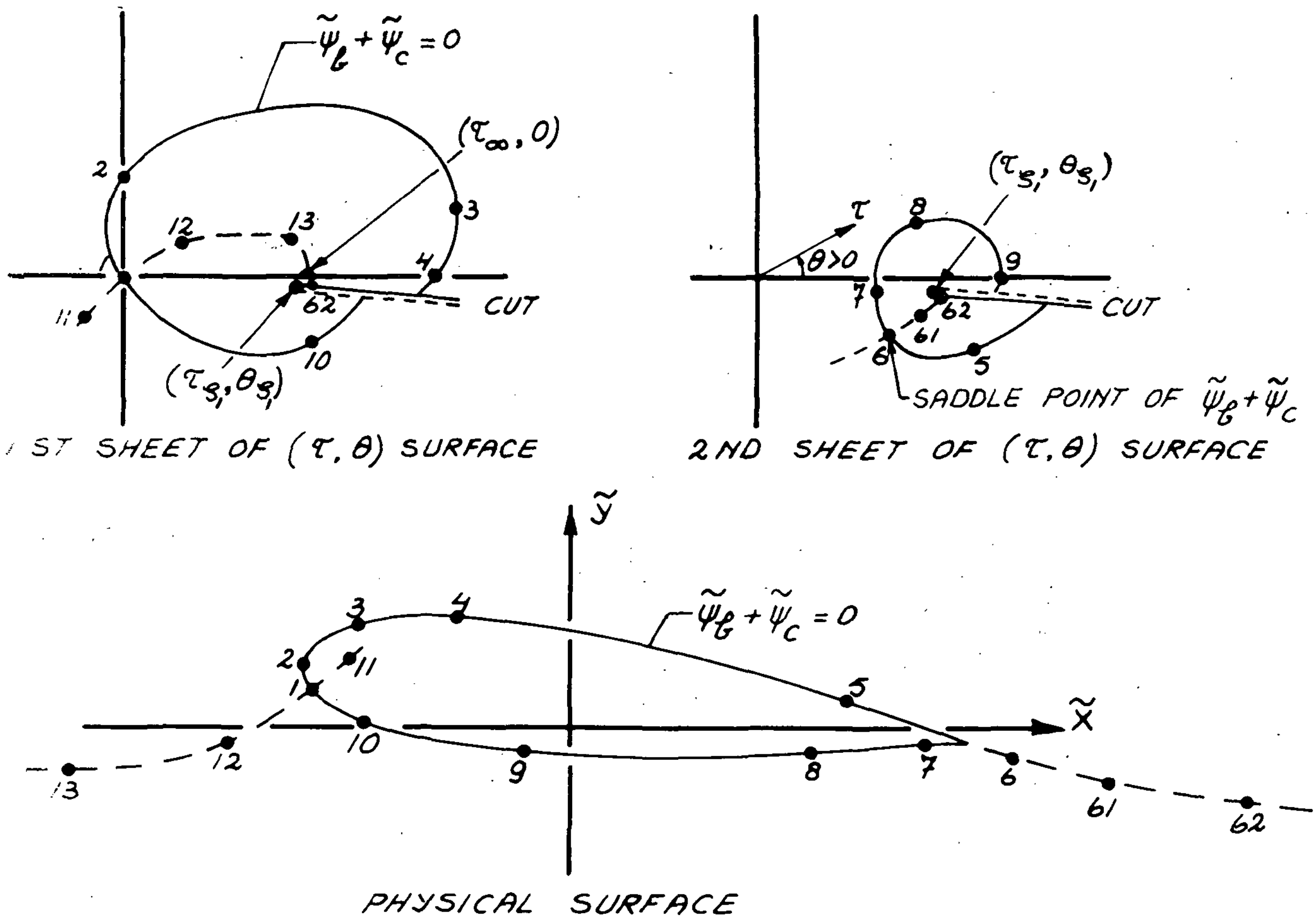


FIG. 2
RELATION BETWEEN HODOGRAPH AND PHYSICAL SURFACES

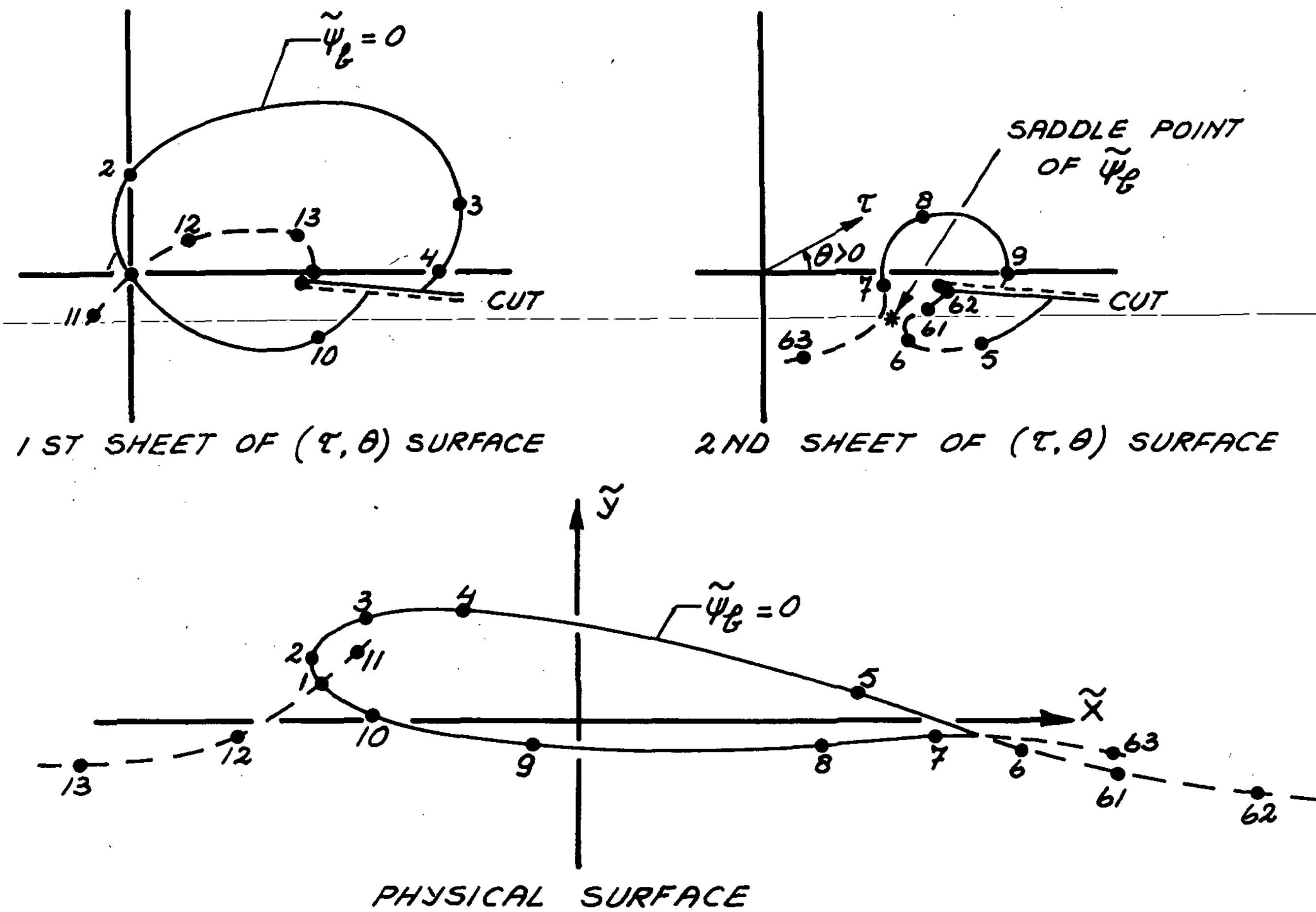
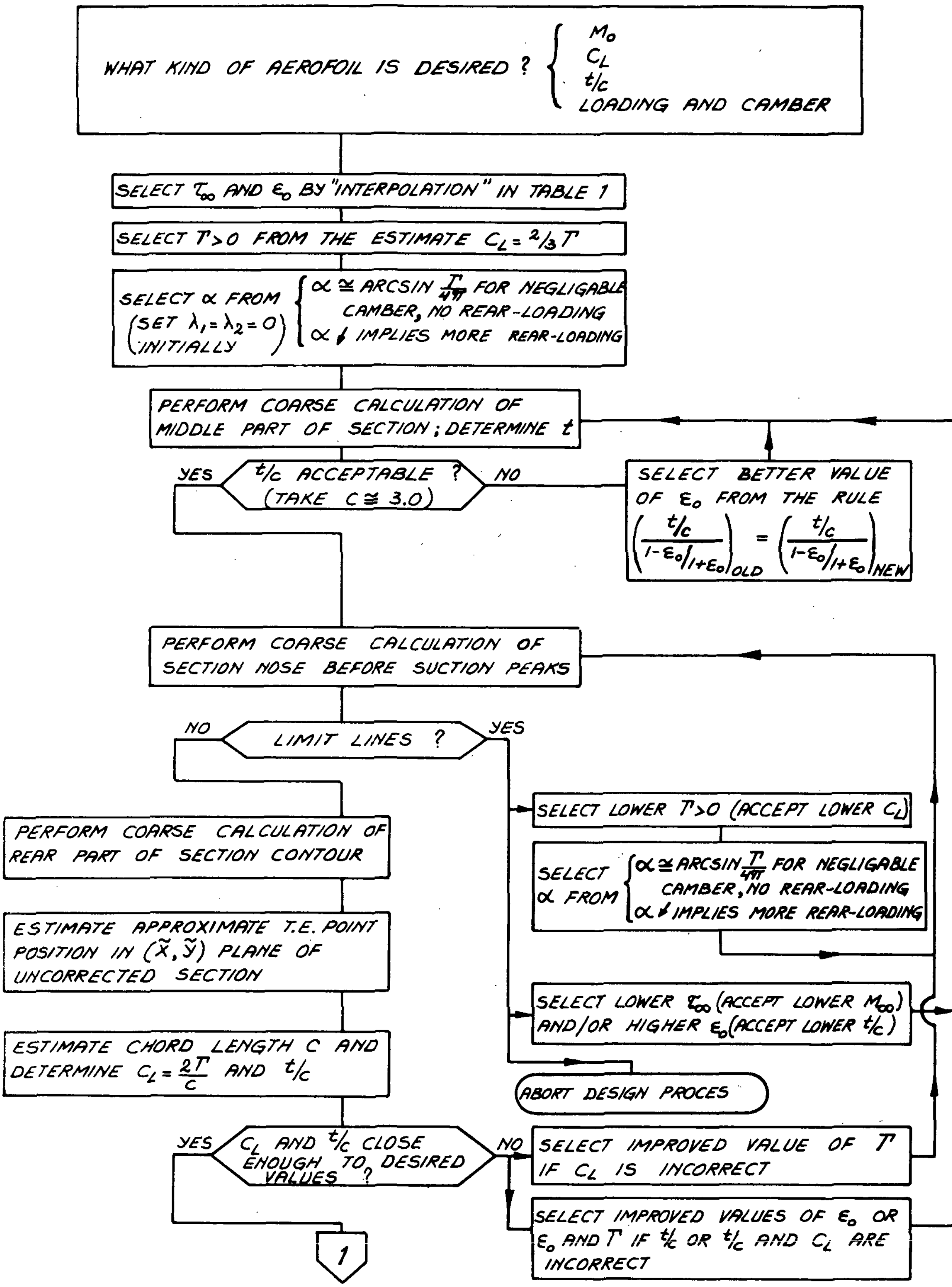
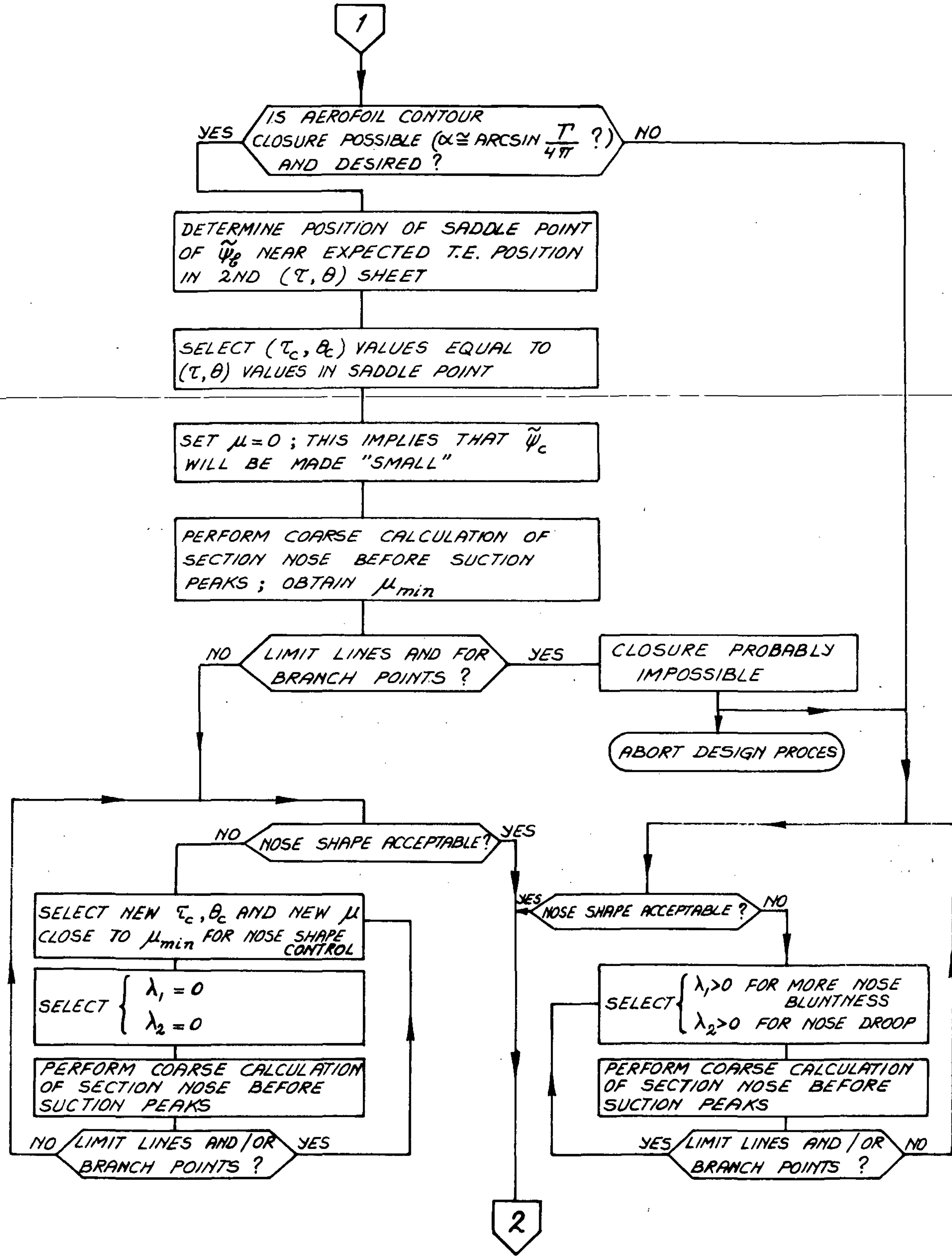


FIG. 3
<u>RELATION BETWEEN HODOGRAPH AND PHYSICAL SURFACES</u>
<u>FOR UNCORRECTED AEROFOILS</u>


WHAT KIND OF AEROFOIL IS DESIRED ?
M_0
C_L
t/c
LOADING AND CAMBER
SELECT τ_∞ AND ϵ_0 BY "INTERPOLATION" IN TABLE 1
SELECT $\Gamma > 0$ FROM THE ESTIMATE $C_L = 2/3\,\Gamma$
SELECT α FROM (SET $\lambda_1 = \lambda_2 = 0$ INITIALLY)
$\alpha \cong$ ARCSIN $\frac{\Gamma}{4\pi}$ FOR NEGLIGABLE CAMBER, NO REAR-LOADING
$\alpha \downarrow$ IMPLIES MORE REAR-LOADING
PERFORM COARSE CALCULATION OF MIDDLE PART OF SECTION; DETERMINE t
YES
t/c ACCEPTABLE ? (TAKE $c \cong 3.0$)
NO
SELECT BETTER VALUE OF ϵ_0 FROM THE RULE
$\left(\frac{t/c}{1-\epsilon_0/1+\epsilon_0}\right)_{OLD} = \left(\frac{t/c}{1-\epsilon_0/1+\epsilon_0}\right)_{NEW}$
PERFORM COARSE CALCULATION OF SECTION NOSE BEFORE SUCTION PEAKS
NO
LIMIT LINES ?
YES
SELECT LOWER $\Gamma > 0$ (ACCEPT LOWER C_L)
SELECT α FROM
$\alpha \cong$ ARCSIN $\frac{\Gamma}{4\pi}$ FOR NEGLIGABLE CAMBER, NO REAR-LOADING
$\alpha \downarrow$ IMPLIES MORE REAR-LOADING
PERFORM COARSE CALCULATION OF REAR PART OF SECTION CONTOUR
ESTIMATE APPROXIMATE T.E. POINT POSITION IN $(\tilde{x}, \tilde{y})$ PLANE OF UNCORRECTED SECTION
SELECT LOWER τ_∞ (ACCEPT LOWER M_∞) AND/OR HIGHER ϵ_0 (ACCEPT LOWER t/c)
ESTIMATE CHORD LENGTH c AND DETERMINE $C_L = \frac{2\Gamma}{c}$ AND t/c
ABORT DESIGN PROCES
YES
C_L AND t/c CLOSE ENOUGH TO DESIRED VALUES ?
NO
SELECT IMPROVED VALUE OF Γ IF C_L IS INCORRECT
SELECT IMPROVED VALUES OF ϵ_0 OR ϵ_0 AND Γ IF t/c OR t/c AND C_L ARE INCORRECT
1


1
IS AEROFOIL CONTOUR CLOSURE POSSIBLE ($\alpha \cong$ ARCSIN $\frac{\Gamma}{4\pi}$?) AND DESIRED ?
YES
NO
DETERMINE POSITION OF SADDLE POINT OF $\tilde{\psi}_\ell$ NEAR EXPECTED T.E. POSITION IN 2ND (τ, θ) SHEET
SELECT (τ_c, θ_c) VALUES EQUAL TO (τ, θ) VALUES IN SADDLE POINT
SET $\mu = 0$; THIS IMPLIES THAT $\tilde{\psi}_c$ WILL BE MADE "SMALL"
PERFORM COARSE CALCULATION OF SECTION NOSE BEFORE SUCTION PEAKS ; OBTAIN μ_{min}
NO
LIMIT LINES AND FOR BRANCH POINTS ?
YES
CLOSURE PROBABLY IMPOSSIBLE
ABORT DESIGN PROCES
NO
NOSE SHAPE ACCEPTABLE?
YES
SELECT NEW τ_c, θ_c AND NEW μ CLOSE TO μ_{min} FOR NOSE SHAPE CONTROL
SELECT $\begin{cases} \lambda_1 = 0 \\ \lambda_2 = 0 \end{cases}$
PERFORM COARSE CALCULATION OF SECTION NOSE BEFORE SUCTION PEAKS
NO
LIMIT LINES AND/OR BRANCH POINTS ?
YES
YES
NOSE SHAPE ACCEPTABLE ?
NO
SELECT $\lambda_1 > 0$ FOR MORE NOSE BLUNTNESS, $\lambda_2 > 0$ FOR NOSE DROOP
PERFORM COARSE CALCULATION OF SECTION NOSE BEFORE SUCTION PEAKS
YES
LIMIT LINES AND/OR BRANCH POINTS ?
NO
2

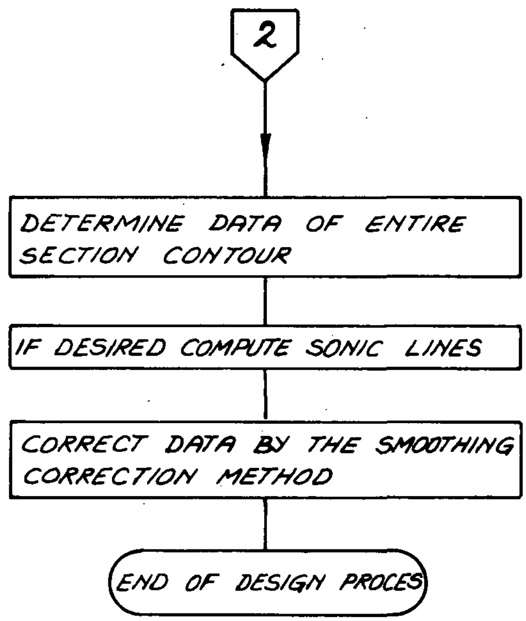


FIG. 4
FLOW CHART OF THE DESIGN PROCES FIXING THE NINE PARAMETERS

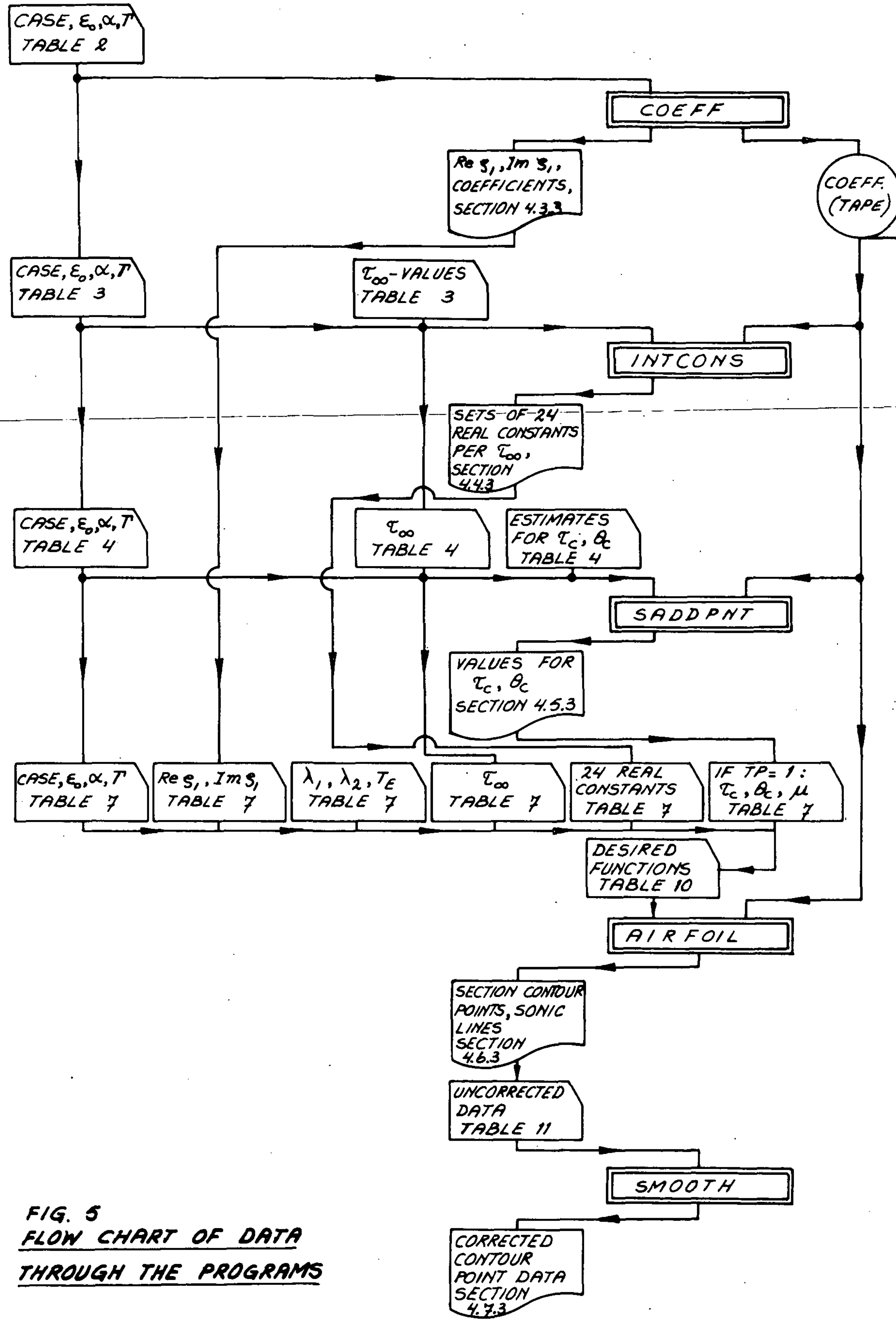


FIG. 5
FLOW CHART OF DATA THROUGH THE PROGRAMS

```
CASE              EPSZERO           ALFA              GAMMA             RE ZETA1          IM ZETA1          RE ZETA2
IM ZETA2          ABSZETA1          ABSZETA2          RE Z1 DIV Z2      IM Z1 DIV Z2      RE EPS            IM EPS
+1.8000000000# +1 +7.1000000000# -1 +4.5000000000# -2 +7.5000000000# -1 +9.8863112240# -1 -2.3226491951# -3 +1.4141192058# +0
-1.2426685672# -1 +9.8863385077# -1 +1.4195687302# +0 +6.9390034233# -1 +5.9334577226# -2 +7.0712644044# -1 +6.3813769931# -2

TOLERANCE 1#-10

HIGHEST SUBSCRIPT OF COEFFICIENTS 100

NUMBERS OF TERMS IN POWER SERIES

  164

 1795

  166

 2009

  165

  165

 1796

  166

  167

 2009

I = 1      K =  1

+0.0000000000000#+000  +0.0000000000000#+000
-6.1026468389320#-003  -2.2812188489718#-004
+1.2540791396711#-002  -9.8961229657548#-003
+2.6269451136262#-002  -1.5517530581701#-002
+3.6161536116631#-002  -1.8490945989969#-002
+4.3092647983731#-002  -1.9772715422159#-002
+4.7762676813075#-002  -2.0008697266284#-002
+5.0723730902516#-002  -1.9628668150989#-002
+5.2406444815774#-002  -1.8913463674781#-002
+5.3143379317007#-002  -1.8042362315068#-002
+5.3189057353379#-002  -1.7126254437059#-002
+5.2736656688616#-002  -1.6230656983713#-002
+5.1931630942217#-002  -1.5391527929747#-002
+5.0882642715929#-002  -1.4626014529952#-002
+4.9670221804946#-002  -1.3939666143749#-002
+4.8353544716914#-002  -1.3331201834804#-002
+4.6975691841884#-002  -1.2795603421149#-002
+4.5567689627233#+002  -1.2326074538168#-002
+4.4151595142954#-002  -1.1915241718423#-002
+4.2742833855830#-002  -1.1555856658619#-
+4.1351960316717#-002  -1.12411765
+3.9985976459494#-002  -1.09
+3.8649313179434#-002
+3.7344557262028#
+3.6072986
+3.4



                                   06#-002
                                 8804112#-002
                               454755142837#-002
                     01  -3.0892859504358#-002
                 04#-001  -3.0343389606803#-002
            3040542#-001  -2.9805978688960#-002
  .5388767994906#-001  -2.9280274689517#-002
-5.4876094064270#-001  -2.8765939479755#-002
-5.4372918838042#-001  -2.8262648145629#-002
-5.3878962165403#-001  -2.7770088315378#-002
-5.3393955494712#-001  -2.7287959529437#-002
-5.2917641255917#-001  -2.6815972649271#-002
-5.2449772283138#-001  -2.6353849302517#-002
-5.1990111273967#-001  -2.5901321361603#-002
-5.1538430282913#-001  -2.5458130453533#-002
-5.1094510246829#-001  -2.5024027498820#-002
-5.0658140539082#-001  -2.4598772277069#-002
-5.0229118551374#-001  -2.4182133017871#-002
-4.9807249300531#-001  -2.3773886014925#-002
-4.9392345058898#-001  -2.3373815262004#-002
-4.8984225006455#-001  -2.2981712109217#-002
-4.8582714903360#-001  -2.2597374938289#-002
-4.8187646781231#-001  -2.2220608855507#-002
-4.7798858652152#-001  -2.1851225401292#-002
-4.7416194234116#-001  -2.1489042275292#-002
-4.7039502692008#-001  -2.1133883076091#-002
-4.6668638392419#-001  -2.0785577054318#-002
-4.6303460672626#-001  -2.0443958878864#-002
-4.5943833621361#-001  -2.0108868414756#-002
```

FIG. 6

EXAMPLE OF OUTPUT OF COEFFT

```
CASE               EPSZERO            ALFA               GAMMA              RE ZETA1           IM ZETA1           RE ZETA2
IM ZETA2           ABSZETA1           ABSZETA2           RE Z1 DIV Z2       IM Z1 DIV Z2       RE EPS             IM EPS
+1.8000000000≠ +1  +7.1000000000≠ -1  +4.5000000000≠ -2  +7.5000000000≠ -1  +9.8863112240≠ -1  -2.3226491951≠ -3  +1.4141192058≠ +0
-1.2426685672≠ -1  +9.8863385077≠ -1  +1.4195687302≠ +0  +6.9390034233≠ -1  +5.9334577226≠ -2  +7.0712644044≠ -1  +6.3813769931≠ -2

NUMBERS OF TERMS IN POWER SERIES

 513

1074

TAU1=0.0750

 +1.489734   +0.043930   -1.489734   -0.043930
 -0.030878   +0.826031   +0.030878   -0.826031
 -0.030878   +0.838718   +0.030878   -0.813344

 -1.489734   -0.043930   +1.489734   +0.043930
 -0.030878   +0.826031   +0.030878   -0.826031
 -0.030878   +0.838718   +0.030878   -0.813344

TAU1=0.0758

 +1.490737   +0.043863   -1.490737   -0.043863
 -0.031151   +0.833329   +0.031151   -0.833329
 -0.031151   +0.845736   +0.031151   -0.820923

 -1.490737   -0.043863   +1.490737   +0.043863
 -0.031151   +0.833329   +0.031151   -0.833329
 -0.031151   +0.845736   +0.031151   -0.820923

TAU1=0.0767

 +1.491787   +0.043799   -1.491787   -0.043799
 -0.031422   +0.840589   +0.031422   -0.840589
 -0.031422   +0.852719   +0.031422   -0.828459

 -1.491787   -0.043799   +1.491787   +0.043799
 -0.031422   +0.840589   +0.031422   -0.840589
 -0.031422   +0.852719   +0.031422   -0.828459

TAU1=0.0775

 +1.492885   +0.043738   -1.492885   -0.043738
 -0.031692   +0.847810   +0.031692   -0.847810
 -0.031692   +0.859666   +0.031692   -0.835953

 -1.492885   -0.043738   +1.492885   +0.043738
 -0.031692   +0.847810   +0.031692   -0.847810
 -0.031692   +0.859666   +0.031692   -0.835953

TAU1=0.0783

 +1.494029   +0.043678   -1.494029   -0.043678
 -0.031960   +0.854992   +0.031960   -0.854992
 -0.031960   +0.866579   +0.031960   -0.843405

 -1.494029   -0.043678   +1.494029   +0.043678
 -0.031960   +0.854992   +0.031960   -0.854992
 -0.031960   +0.866579   +0.031960   -0.843405

TAU1=0.0792

 +1.495219   +0.043621   -1.495219   -0.043621
 -0.032227   +0.862136   +0.032227   -0.862136
 -0.032227   +0.873456   +0.032227   -0.850815

 -1.495219   -0.043621   +1.495219   +0.043621
 -0.032227   +0.862136   +0.032227   -0.862136
 -0.032227   +0.873456   +0.032227   -0.850815

TAU1=0.0800

 +1.496456   +0.043566   -1.496456   -0.043566
 -0.032493   +0.869241   +0.032493   -0.869241
 -0.032493   +0.880299   +0.032493   -0.858184

 -1.496456   -0.043566   +1.496456   +0.043566
 -0.032493   +0.869241   +0.032493   -0.869241
 -0.032493   +0.880299   +0.032493   -0.858184
```

FIG. 7

EXAMPLE OF OUTPUT OF INTCONS

```
CASE          18
EPSILON(0)=0.710000
ALFA      =0.045000
GAMMA     =0.750000

TAU 1 =0.077500
```

TAU(C)		THETA(C)		PSI		DPSI/DTAU		DPSI/DTHETA		
+4.650000100'	-2	-3.300000000'	+0	-1.509974184'	-2	+2.414259265'	-1	-4.535078159'	-2	STEP 1
+4.650000100'	-2	-3.800000000'	+0	-1.477725776'	-2	+8.532782331'	-1	-2.717375376'	-2	
+4.650000100'	-2	-2.800000000'	+0	-1.554115277'	-2	-4.001338056'	-1	-5.396004725'	-2	
+5.150000100'	-2	-3.300000000'	+0	-1.163153957'	-2	+1.318132882'	+0	-5.496345307'	-1	
+4.150000100'	-2	-3.300000000'	+0	-1.509377224'	-2	-1.768594275'	-1	+2.182432770'	-1	
+4.586848652'	-2	-3.182708165'	+0	-1.523300173'	-2	+2.754594976'	-2	-4.104430697'	-3	STEP 2
+4.586848652'	-2	-3.065416330'	+0	-1.524371442'	-2	-1.103441759'	-1	-6.096283701'	-3	
+4.586848652'	-2	-3.300000000'	+0	-1.522686679'	-2	+1.641817040'	-1	-1.641443737'	-3	
+4.523697205'	-2	-3.182708165'	+0	-1.523150302'	-2	-3.014841733'	-2	+3.652174626'	-2	
+4.650000100'	-2	-3.182708165'	+0	-1.519572826'	-2	+9.307143098'	-2	-4.825771132'	-2	
+4.580224253'	-2	-3.164692427'	+0	-1.523455757'	-2	+2.241396158'	-4	-1.101289147'	-5	STEP 3
+4.580224253'	-2	-3.146676688'	+0	-1.523464080'	-2	-2.072322096'	-2	-3.348330217'	-4	
+4.580224253'	-2	-3.182708165'	+0	-1.523457702'	-2	+2.114270526'	-2	+3.187978276'	-4	
+4.573599853'	-2	-3.164692427'	+0	-1.523433237'	-2	-5.912910031'	-3	+4.380275734'	-3	
+4.586848652'	-2	-3.164692427'	+0	-1.523437387'	-2	+6.443779659'	-3	-4.439953441'	-3	
+4.580202921'	-2	-3.164516646'	+0	-1.523455777'	-2	-2.756908657'	-8	-1.059754068'	-5	LAST STEP

FIG. 8

EXAMPLE OF OUTPUT OF SADDPNT

THE COMPUTATION OF A QUASI-ELLIPTICAL AEROFOIL
IN A CIRCULATORY TRANSONIC POTENTIAL FLOW
BY USING LIGHTHILLS 2ND INTEGRAL OPERATOR

EPSILON(0)=0.7100   ALFA=+0.045000   GAMMA=0.750000

TAU(1)=0.0775   TAU(ZETA1)=0.0752

CASE 18   M-INF = 0.6481

LAMBDA1 = +0.00   LAMBDA2 = +0.00

CORRECTION FUNCTION QUANTITIES..

TAU(C)=0.045800   THETA(C)=-3.220020

-.1036≠+1	-.7343≠-1	-.7307≠-1	+.6473≠+0	+0.01523510
+.1230≠+2	-.1802≠+1	-.1406≠+1	+.1246≠+2	-0.06414442
-.1726≠+0	-.1366≠+1	-.1295≠+1	-.1461≠+0	+0.00118564
-.1109≠-1	-.3381≠-4	-.5489≠-4	+.5783≠-2	

INTEGRATION CONSTANTS..

J=1	+1.49286	+0.03697	-1.49286	-0.03697
J=2	-0.01220	+0.84777	+0.01220	-0.84777
J=3	-0.01220	+0.85963	+0.01220	-0.83592

J=1	-1.49286	-0.03697	+1.49286	+0.03697
J=2	-0.01220	+0.84777	+0.01220	-0.84777
J=3	-0.01220	+0.85963	+0.01220	-0.83592

STAGNATION POINT

THETA	X	Y	1/R	CP
-1.25208	-1.49286	+0.03697	.65289≠ +2	+1.1095

MU2=+7.4055700000018≠-001

FIG. 9   PAGE 1
EXAMPLE OF OUTPUT OF AIRFOIL

```
TAU=0.120000     M=0.8257     CP=-0.5175

TOL=.000010      J=3


                           UPPER REAR PART

   THETA       PSI          X          Y     DPSI/DTAU DPSI/DTHETA    DET J
                       DX/DTHETA DY/DTHETA                             1/R

 -12.0000   -0.32364
 -10.0000   -0.15668
  -8.0000   -0.06467
  -6.0000   +0.14028
  -7.3689   -0.01030
  -7.2493   -0.00018
  -7.2472   +0.00001
  -7.2478   -0.00004
  -7.2472   +0.00001   +0.39838   +0.16256 -19.78572   +5.16621   -.40760≠ -1

 -0.12649              -4.49018   +6.33215                        +.13828≠ +0

  K   MAX     13   31    44   33    28   33    44   47    28   47

  KC  MAX      9   16     4   17     9   17     9   19     9   20




TAU=0.130000     M=0.8644     CP=-0.6306

TOL=.000010      J=3


                           UPPER REAR PART

   THETA       PSI          X          Y     DPSI/DTAU DPSI/DTHETA    DET J
                       DX/DTHETA DY/DTHETA                             1/R

  -7.2472   -0.19344
  -5.2472   -0.03587
  -3.2472   +0.13510
  -4.8276   -0.00017
  -4.8256   +0.00000
  -4.8263   -0.00005
  -4.8256   +0.00000   +0.08524   +0.19503 -22.87835   +4.90903   -.33812≠ -1

 -0.08422              -6.03077   +5.89703                        +.13113≠ +0

  K   MAX     44   26    44   35    44   34    44   41    44   45

  KC  MAX      9   15     5   15     4   17     4   22     4   18
```

FIG. 9   PAGE 2

EXAMPLE OF OUTPUT OF AIRFOIL

```
TAU=0.140000     M=0.9022     CP=-0.7404

TOL=.000010      J=3
```

UPPER REAR PART

```
   THETA       PSI          X           Y      DPSI/DTAU DPSI/DTHETA     DET J
                      DX/DTHETA DY/DTHETA                                1/R

 -4.8256   -0.21878
 -2.8256   -0.06777
 -0.8256   +0.07029
 -1.8439   +0.00422
 -1.9015   +0.00026
 -1.9053   -0.00000
 -1.9047   +0.00004
 -1.9053   -0.00000   -0.30137   +0.21724 -22.47466   +4.03032   -.36017≠ -1

-0.03325              -6.67727   +4.59654                         +.13610≠ +0

  K  MAX     38   28    38   29    38   28    38   46    27   43

  KC MAX      9   12     4   13     5   13     5   15     4   15
```

```
TAU=0.145000     M=0.9208     CP=-0.7942

TOL=.000010      J=3
```

UPPER REAR PART

```
   THETA       PSI          X           Y      DPSI/DTAU DPSI/DTHETA     DET J
                      DX/DTHETA DY/DTHETA                                1/R

 -0.4451   -0.01331
 +0.5549   +0.04062
 -0.1983   +0.00099
 -0.2154   +0.00001
 -0.2160   -0.00002
 -0.2154   +0.00001   -0.50797   +0.22179 -20.27334   +3.26945   -.44194≠ -1

-0.00376              -6.34546   +3.55999                         +.15020≠ +0

  K  MAX     38   25    27   35    38   25    38   43    38   48

  KC MAX      9   13     4   14     5   14     9   16     4   14
```

FIG. 9 PAGE 3
EXAMPLE OF OUTPUT OF AIRFOIL

```
TAU=0.150000     M=0.9393     CP=-0.8472

TOL=.000010      J=3


                              UPPER REAR PART

   THETA       PSI          X           Y      DPSI/DTAU DPSI/DTHETA    DET J
                        DX/DTHETA DY/DTHETA                             1/R

 +1.4745  -0.00917
 +2.4745  +0.02792
 +1.7216  +0.00114
 +1.6944  +0.00003
 +1.6937  -0.00000
 +1.6937  -0.00000   -0.71299   +0.21906 -16.93044   +2.33106   -.62873≠ -1

+0.02956             -5.55279   +2.35234                        +.17804≠ +0

 K   MAX    38  41    38  36     27  44    38  55     38  50

 KC  MAX     9  12    10  13      5  13     4  18     10  16



TAU=0.155000     M=0.9577     CP=-0.8994

TOL=.000010      J=3


                              UPPER REAR PART

   THETA       PSI          X           Y      DPSI/DTAU DPSI/DTHETA    DET J
                        DX/DTHETA DY/DTHETA                             1/R

 +3.6028  -0.01177
 +4.6028  +0.01122
 +4.1149  +0.00121
 +4.0558  -0.00102
 +4.0829  +0.00048
 +4.0742  +0.00029
 +4.0650  -0.00082
 +4.0718  +0.00023
 +4.0684  -0.00074
 +4.0710  +0.00021
 +4.0697  -0.00071
 +4.0704  +0.00020
 +4.0704  +0.00020   -0.91776   +0.20886 -12.73044   +1.31921   -.10963≠ +0

+0.07104             -4.34172   +1.11584                        +.23305≠ +0

 K   MAX    38  31    27  41     27  39    27  47     38  52

 KC  MAX     5  14     4  15      5  16     4  20      4  15
```

FIG. 9 PAGE 4

EXAMPLE OF OUTPUT OF AIRFOIL

```
TAU=0.160000     M=0.9759     CP=-0.9508

TOL=.000010      J=3


                              UPPER REAR PART

   THETA       PSI          X            Y      DPSI/DTAU DPSI/DTHETA    DET J
                       DX/DTHETA DY/DTHETA                                1/R

  +6.4471   -0.01256
  +7.4471   -0.00193
  +8.4471   +0.00484
  +7.7318   +0.00033
  +7.6902   +0.00002
  +7.6873   -0.00000
  +7.6880   +0.00000
  +7.6873   -0.00000   -1.13307   +0.18741   -7.71791   +0.43662   -.29180≠ +0

 +0.13417              -2.69689   +0.11012                         +.37567≠ +0

  K  MAX     38  28     27  46     27  39     27  49     38  43

  KC MAX      5  14      4  15      5  16      9  21      4  15
```

```
TAU=0.165000     M=0.9940     CP=-1.0014

TOL=.000010      J=3


                              UPPER REAR PART

   THETA       PSI          X            Y      DPSI/DTAU DPSI/DTHETA    DET J
                       DX/DTHETA DY/DTHETA                                1/R

 +11.3043   -0.01106
 +12.3043   -0.00829
 +13.3043   -0.00602
 +14.3043   -0.00502
 +15.3043   -0.00404
 +16.3043   -0.00364
 +17.3043   -0.00333
 +18.3043   -0.00316
 +19.3043   -0.00313
 +18.8043   -0.00313
 +19.0543   -0.00313
```

FIG. 9 PAGE 5
EXAMPLE OF OUTPUT OF AIRFOIL

TAU=0.045800 M=0.4899 CP=+0.4269

TOL=.000010 J=1

TAIL POINT

THETA	PSI	X DX/DTHETA	Y DY/DTHETA	DPSI/DTAU	DPSI/DTHETA	DET J 1/R	
-3.2200	+0.00000	+1.61909	-0.07062	+0.00000	-0.00000	+.00000≠	+0
-0.05620		+0.00000	-0.00000			+.00000≠	+0
K MAX	32 22	17 24	17 23	32 36	32 34		
KC MAX	3 12	8 19	3 19	3 20	8 17		

TAU=0.166667 M=1.0000 CP=-1.0182

TOL=.000010 J=3

UPPER SONIC LINE

THETA	PSI	X	Y
+0.5000	-0.33671	-0.60175	-0.14131
+1.5000	-0.26939	-0.69651	-0.07058
+3.0000	-0.18586	-0.82692	+0.01430
+5.0000	-0.10499	-0.97470	+0.08827
+8.0000	-0.04571	-1.14059	+0.13689
+12.0000	-0.01578	-1.27759	+0.14555
+16.0000	-0.00768	-1.35458	+0.13579
+20.0000	-0.00536	-1.39855	+0.12399
+25.0000	-0.00255	-1.42923	+0.11136
+30.0000	-0.01986	-1.44596	+0.10134

FIG. 9 PAGE 6
EXAMPLE OF OUTPUT OF AIRFOIL

RESULTS OF PROGRAM T 32

DATA INPUT TAPE NUMBER A*+1.601000000000'+003*

+0	-1.6399500'	+0	+0.0000000'	+0	+1.4400000'	+0	+4.1021800'	+1
+1	-1.6375300'	+0	+8.5900000'	-3	+1.1475400'	+0	+2.8340000'	+1
+2	-1.6351700'	+0	+1.3000000'	-2	+1.0219600'	+0	+2.2239000'	+1
+3	-1.6325200'	+0	+1.7020000'	-2	+9.2394000'	-1	+1.7583000'	+1
+4	-1.6293700'	+0	+2.0760000'	-2	+8.4722000'	-1	+1.4626000'	+1
+5	-1.6257500'	+0	+2.4580000'	-2	+7.7744000'	-1	+1.2009000'	+1
+6	-1.6215200'	+0	+2.8470000'	-2	+7.1479000'	-1	+9.9302000'	+0
+7	-1.6165900'	+0	+3.2590000'	-2	+6.5629000'	-1	+8.1719000'	+0
+8	-1.6108100'	+0	+3.6700000'	-2	+6.0388000'	-1	+6.8004000'	+0
+9	-1.6040100'	+0	+4.1110000'	-2	+5.5384000'	-1	+5.6147000'	+0
+10	-1.5959700'	+0	+4.5860000'	-2	+5.0628000'	-1	+4.6132000'	+0
+11	-1.5865000'	+0	+5.0950000'	-2	+4.6171000'	-1	+3.7878000'	+0
+12	-1.5743500'	+0	+5.6300000'	-2	+4.2004000'	-1	+3.1126000'	+0
+13	-1.5613600'	+0	+6.1850000'	-2	+3.8026000'	-1	+2.5288000'	+0
+14	-1.5468100'	+0	+6.7050000'	-2	+3.4426000'	-1	+2.1418000'	+0
+15	-1.5301500'	+0	+7.3130000'	-2	+3.0909000'	-1	+1.8132000'	+0
+16	-1.5137000'	+0	+7.6970000'	-2	+2.8038000'	-1	+1.6349000'	+0
+17	-1.4967600'	+0	+8.1600000'	-2	+2.5303000'	-1	+1.5487000'	+0
+18	-1.4800900'	+0	+8.7410000'	-2	+2.2756000'	-1	+1.5974000'	+0
+19	-1.4704400'	+0	+9.0010000'	-2	+2.0778000'	-1	+1.9832000'	+0
+20	-1.4622700'	+0	+9.0750000'	-2	+1.9003000'	-1	+2.8911000'	+0
+21	-1.4574500'	+0	+9.0600000'	-2	+1.7371000'	-1	+2.7348000'	+0
+22	-1.3942200'	+0	+9.1770000'	-2	+1.2611000'	-1	+4.0928000'	-1
+23	-1.2858800'	+0	+1.1019000'	-1	+9.0890000'	-2	+2.4922000'	-1
+24	-1.1392400'	+0	+1.2704000'	-1	+6.3920000'	-2	+1.6921000'	-1
+25	-9.4031000'	-1	+1.3385000'	-1	+3.5760000'	-2	+1.0172000'	-1
+26	-5.3171000'	-1	+1.4045000'	-1	+3.5300000'	-3	+6.6507000'	-2
+27	-5.6950000'	-2	+1.3400000'	-1	-2.8190000'	-2	+7.2089000'	-2
+28	+2.3045000'	-1	+1.2327000'	-1	-5.0580000'	-2	+8.2760000'	-2
+29	+4.3327000'	-1	+1.0822000'	-1	-6.7450000'	-2	+8.0466000'	-2
+30	+6.0441000'	-1	+9.8040000'	-2	-8.0050000'	-2	+6.5220000'	-2
+31	+7.6454000'	-1	+8.5560000'	-2	-8.8460000'	-2	+3.7081000'	-2
+32	+9.2548000'	-1	+7.0300000'	-2	-9.0680000'	-2	+1.2599000'	-2
+33	+1.1470000'	+0	+4.8000000'	-2	-9.1000000'	-2	+0.0000000'	+0
I	X_i		Y_i		θ_i		$(1/R)_i$	

FIG. 10 PAGE 1

EXAMPLE OF OUTPUT OF SMOOTH

MODIFIED INPUT DATA

I	X_i		Y_i		Y_i'		Y_i''	
+0	-1.6399500'	+0	+0.0000000'	+0	+7.6018261'	+0	-1.8490348'	+4
+1	-1.6375300'	+0	+8.5900000'	-3	+2.2198348'	+0	-4.0900205'	+2
+2	-1.6351700'	+0	+1.3000000'	-2	+1.6353089'	+0	-1.5662666'	+2
+3	-1.6325200'	+0	+1.7020000'	-2	+1.3240548'	+0	-8.0321336'	+1
+4	-1.6293700'	+0	+2.0760000'	-2	+1.1319705'	+0	-5.0398256'	+1
+5	-1.6257500'	+0	+2.4580000'	-2	+9.8420901'	-1	-3.3171519'	+1
+6	-1.6215200'	+0	+2.8470000'	-2	+8.6789196'	-1	-2.3052537'	+1
+7	-1.6165900'	+0	+3.2590000'	-2	+7.7017728'	-1	-1.6433023'	+1
+8	-1.6108100'	+0	+3.6700000'	-2	+6.8984800'	-1	-1.2193140'	+1
+9	-1.6040100'	+0	+4.1110000'	-2	+6.1840116'	-1	-9.1261282'	+0
+10	-1.5959700'	+0	+4.5860000'	-2	+5.5448491'	-1	-6.8967053'	+0
+11	-1.5865000'	+0	+5.0950000'	-2	+4.9758033'	-1	-5.2782855'	+0
+12	-1.5743500'	+0	+5.6300000'	-2	+4.4662052'	-1	-4.0889080'	+0
+13	-1.5613600'	+0	+6.1850000'	-2	+3.9971423'	-1	-3.1584435'	+0
+14	-1.5468100'	+0	+6.7050000'	-2	+3.5853719'	-1	-2.5677900'	+0
+15	-1.5301500'	+0	+7.3130000'	-2	+3.1932442'	-1	-2.0974868'	+0
+16	-1.5137000'	+0	+7.6970000'	-2	+2.8796579'	-1	-1.8424187'	+0
+17	-1.4967600'	+0	+8.1600000'	-2	+2.5857198'	-1	-1.7065858'	+0
+18	-1.4800900'	+0	+8.7410000'	-2	+2.3157106'	-1	-1.7275987'	+0
+19	-1.4704400'	+0	+9.0010000'	-2	+2.1082268'	-1	-2.1168770'	+0
+20	-1.4622700'	+0	+9.0750000'	-2	+1.9235095'	-1	-3.0530263'	+0
+21	-1.4574500'	+0	+9.0600000'	-2	+1.7547859'	-1	-2.8620854'	+0
+22	-1.3942200'	+0	+9.1770000'	-2	+1.2678282'	-1	-4.1918763'	-1
+23	-1.2858800'	+0	+1.1019000'	-1	+9.1141110'	-2	-2.5233173'	-1
+24	-1.1392400'	+0	+1.2704000'	-1	+6.4007197'	-2	-1.7025092'	-1
+25	-9.4031000'	-1	+1.3385000'	-1	+3.5775251'	-2	-1.0191534'	-1
+26	-5.3171000'	-1	+1.4045000'	-1	+3.5300147'	-3	-6.6508243'	-2
+27	-5.6950000'	-2	+1.3400000'	-1	-2.8197470'	-2	-7.2174994'	-2
+28	+2.3045000'	-1	+1.2327000'	-1	-5.0623178'	-2	-8.3078338'	-2
+29	+4.3327000'	-1	+1.0822000'	-1	-6.7552474'	-2	-8.1017418'	-2
+30	+6.0441000'	-1	+9.8040000'	-2	-8.0221426'	-2	-6.5850595'	-2
+31	+7.6454000'	-1	+8.5560000'	-2	-8.8691463'	-2	-3.7519388'	-2
+32	+9.2548000'	-1	+7.0300000'	-2	-9.0929370'	-2	-1.2755578'	-2
+33	+1.1470000'	+0	+4.8000000'	-2	-9.1252025'	-2	+0.0000000'	+0

FIG. 10 PAGE 2

EXAMPLE OF OUTPUT OF SMOOTH

```
DATA INPUT TAPE NUMBER B*+1.602000000000'+003*
EPS    =+10.0000000'  -1
NU0    =+10.0000000'  -3
NU1    =+10.0000000'  -2
NU2    =+10.0000000'  -1
MU3    =+10.0000000'  -1
MU4    =+10.0000000'  -1
MU5    =+10.0000000'  -1
NSIGMA= +6
KKMAX = +1
KMAX  =+10
TOL1   =+1.0000000'  -5
TOL2   =+1.0000000'  -5
SIGMA0,SIGMA1,SIGMA2,K
+1.0000000' -30 +1.0000000' -30 +1.0000000' -30   +0
+1.0000000' -30 +1.0000000' -30 +1.0000000' -30   +5
+1.0000000'  -8 +1.0000000'  -6 +1.0000000'  -4  +15
+1.0000000'  -6 +1.0000000'  -6 +1.0000000'  -6  +23
+1.0000000'  -6 +1.0000000'  -6 +1.0000000'  -6  +32
+1.0000000' -30 +1.0000000'  -6 +1.0000000'  -6  +33
RHO3,RHO4,RHO5,K
+1.0000000'  +0 +1.0000000'  +0 +1.0000000'  +0  -1
```

$\} N_\sigma \times \left\{ (\Delta Y)_j^2, (\Delta Y')_j^2, (\Delta Y'')_j^2, j \right\}$

FIG. 10 PAGE 3

EXAMPLE OF OUTPUT OF SMOOTH

```
KKC=  1
K=   +0
E   =+0.0000000'  +0
S   =+3.2999998'  +1
WEIGHT TABLE FOR THE RHO I
I       RHO I
 +0 +1.5167978' -12
 +1 +2.6318035'  -8
 +2 +8.6468692'  -9
 +3 +3.9643051'  -8
 +4 +5.9068438'  -6
 +5 +9.6913410'  -6
 +6 +4.4114499'  -7
 +7 +9.3622449'  -7
 +8 +2.8174829'  -5
 +9 +2.0841564'  -5
+10 +7.2622034'  -6
+11 +2.6636871'  -6
+12 +1.1651879'  -4
+13 +9.6791160'  -6
+14 +9.0293142'  -6
+15 +1.2667598'  -6
+16 +9.3035599'  -2
+17 +6.0142708'  -7
+18 +5.4349358'  -7
+19 +6.0760013'  -8
+20 +3.3658803'  -9
+21 +2.4340303'  -5
+22 +4.5115203'  -4
+23 +2.9790369'  -3
+24 +5.1754407'  -2
+25 +1.6821194'  +1
+26 +6.7154153'  +1
+27 +9.9725589'  +0
+28 +5.0948630'  -2
+29 +3.2570199'  -2
+30 +1.1978289'  -1
+31 +2.6393892'  -1
+32 +1.7293302'  -1
```

```
K  =  +1
EPS   =+.304138420415'  +1
I   = +83
C6   =+.767751483814'  +7
NUMBER OF NON SIGNIFICANT CONTRIBUTIONS IN C6 AND E..  +0
INFINITY NORM OF YCORR    +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.8490348'  +4
INFINITY NORM OF YCORR    +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+2.1459983' -12
NUMBER OF ITERATIONS IN RESIDUAL VECTOR METHOD.. +2
TOLERANCE TESTS ARE SATISFIED
E   =+6.1066405'  -1
S   =+6.5181797'  +0
```

FIG. 10   PAGE 4

EXAMPLE OF OUTPUT OF SMOOTH

```
K  =  +2
EPS   =+.260875572782'  -2
I   = +83
C6   =+.564865335138'  +1
NUMBER OF NON SIGNIFICANT CONTRIBUTIONS IN C6 AND E..  +0
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.8490348'  +4
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+2.0009322'  -9
NUMBER OF ITERATIONS IN RESIDUAL VECTOR METHOD.. +2
TOLERANCE TESTS ARE SATISFIED
E   =+4.6284200'  +0
S   =+5.4711929'  +0


K  =  +3
EPS   =+.145474904605'  -3
I   = +83
C6   =+.313153540813'  -4
NUMBER OF NON SIGNIFICANT CONTRIBUTIONS IN C6 AND E..  +4
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.8490348'  +4
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+9.4806842'  -9
NUMBER OF ITERATIONS IN RESIDUAL VECTOR METHOD.. +2
TOLERANCE TESTS ARE SATISFIED
E   =+1.4664412'  +2
S   =+5.4232079'  +0


K  =  +4
EPS   =+.257024577354'  -3
I   = +83
C6   =+.310227841904'  -5
NUMBER OF NON SIGNIFICANT CONTRIBUTIONS IN C6 AND E..  +8
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.8490348'  +4
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.7791682'  -8
NUMBER OF ITERATIONS IN RESIDUAL VECTOR METHOD.. +2
TOLERANCE TESTS ARE SATISFIED
E   =+6.5517164'  +1
S   =+5.4385281'  +0


K  =  +5
EPS   =+.202885800044'  -3
I   = +83
C6   =+.432477744666'  -5
NUMBER OF NON SIGNIFICANT CONTRIBUTIONS IN C6 AND E..  +9
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.8490348'  +4
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+2.0982066'  -8
INFINITY NORM OF YCORR     +1.8490348'  +4
INFINITY NORM OF IMPROVEMENT VECTOR+1.5144470'  -8
NUMBER OF ITERATIONS IN RESIDUAL VECTOR METHOD.. +3
TOLERANCE TESTS ARE SATISFIED
E   =+9.1827910'  +1
S   =+5.4325453'  +0
SMOOTHING COMPLETED AT K = +5
```

FIG. 10   PAGE 5

EXAMPLE OF OUTPUT OF SMOOTH

RESULTS OF THE SMOOTHING PROCESS

I	X_i	$\hat{Y}_i$	$\hat{Y}'_i$	$\hat{Y}''_i$	$\hat{Y}_i - Y_i$	$\hat{Y}'_i - Y'_i$	$\hat{Y}''_i - Y''_i$
+0	-1.6399500' +0	+3.5864425' -23	+7.6018261' +0	-1.8490348' +4	+3.5864425' -23	+0.0000000' +0	+0.0000000' +0
+1	-1.6375300' +0	+8.5900000' -3	+2.2198348' +0	-4.0900205' +2	+0.0000000' +0	+0.0000000' +0	+0.0000000' +0
+2	-1.6351700' +0	+1.3000000' -2	+1.6353089' +0	-1.5662666' +2	+0.0000000' +0	+0.0000000' +0	+0.0000000' +0
+3	-1.6325200' +0	+1.7020000' -2	+1.3240548' +0	-8.0321336' +1	+0.0000000' +0	+0.0000000' +0	+0.0000000' +0
+4	-1.6293700' +0	+2.0760000' -2	+1.1319705' +0	-5.0398256' +1	+0.0000000' +0	+0.0000000' +0	+0.0000000' +0
+5	-1.6257500' +0	+2.4580000' -2	+9.8420901' -1	-3.3171519' +1	+0.0000000' +0	+0.0000000' +0	+0.0000000' +0
+6	-1.6215200' +0	+2.8476809' -2	+8.6538003' -1	-2.3063347' +1	+6.8087339' -6	-2.5119349' -3	-1.0809435' -2
+7	-1.6165900' +0	+3.2496155' -2	+7.7063531' -1	-1.6432657' +1	-9.3844942' -5	+4.5803419' -4	+3.6579977' -4
+8	-1.6108100' +0	+3.6705818' -2	+6.9008619' -1	-1.2186835' +1	+5.8178070' -6	+2.3818992' -4	+6.3055408' -3
+9	-1.6040100' +0	+4.1140626' -2	+6.1773472' -1	-9.1290648' +0	+3.0626171' -5	-6.6643554' -4	-2.9366476' -3
+10	-1.5959700' +0	+4.5837186' -2	+5.5355283' -1	-6.8984667' +0	-2.2814101' -5	-9.3208652' -4	-1.7613896' -3
+11	-1.5865000' +0	+5.0798944' -2	+4.9689487' -1	-5.2782582' +0	-1.5105558' -4	-6.8546093' -4	+2.7250100' -5
+12	-1.5743500' +0	+5.6506940' -2	+4.4511547' -1	-4.0844850' +0	+2.0694048' -4	-1.5050570' -3	+4.4229756' -3
+13	-1.5613600' +0	+6.1970965' -2	+3.9814811' -1	-3.1620058' +0	+1.2096500' -4	-1.5661209' -3	-3.5623073' -3
+14	-1.5468100' +0	+6.7456729' -2	+3.5735208' -1	-2.5676927' +0	+4.0672922' -4	-1.1851166' -3	+9.7268652' -5
+15	-1.5301500' +0	+7.3079595' -2	+3.1896914' -1	-2.0975350' +0	-5.0404646' -5	-3.5528267' -4	-4.8171733' -5
+16	-1.5137000' +0	+7.8067264' -2	+2.8815726' -1	-1.8389301' +0	+1.0972644' -3	+1.9146662' -4	+3.4886803' -3
+17	-1.4967600' +0	+8.2690959' -2	+2.5809725' -1	-1.7100734' +0	+1.0909586' -3	-4.7473858' -4	-3.4875758' -3
+18	-1.4800900' +0	+8.6767710' -2	+2.3099567' -1	-1.7275989' +0	-6.4228958' -4	-5.7538652' -4	-1.9384179' -7
+19	-1.4704400' +0	+8.8904887' -2	+2.1132000' -1	-2.1168771' +0	-1.1051134' -3	+4.9731701' -4	-6.6058760' -8
+20	-1.4622700' +0	+9.0558387' -2	+1.9219627' -1	-3.0530262' +0	-1.9161326' -4	-1.5467740' -4	+9.1225530' -8
+21	-1.4574500' +0	+9.1450527' -2	+1.7817788' -1	-2.8520720' +0	+8.5052660' -4	+2.6992863' -3	+1.3491836' -5
+22	-1.3942200' +0	+1.0019759' -1	+1.2439401' -1	-4.1917588' -1	+8.4275880' -3	-2.3888088' -3	+1.1747669' -5
+23	-1.2858800' +0	+1.1173546' -1	+9.1669700' -2	-2.5231053' -1	+1.5454641' -3	+5.2858958' -4	+2.1207403' -5
+24	-1.1392400' +0	+1.2298127' -1	+6.3746081' -2	-1.7013311' -1	-4.0587271' -3	-2.6111524' -4	+1.1781699' -4
+25	-9.4031000' -1	+1.3283063' -1	+3.7713487' -2	-9.6835337' -2	-1.0193726' -3	+1.9382362' -3	+5.0800081' -3
+26	-5.3171000' -1	+1.4085696' -1	+3.2701333' -3	-7.1926733' -2	+4.0696209' -4	-2.5988140' -4	-5.4184899' -3
+27	-5.6950000' -2	+1.3427420' -1	-3.1071031' -2	-7.2813504' -2	+2.7420478' -4	-2.8735610' -3	-6.3851037' -4
+28	+2.3045000' -1	+1.2222062' -1	-5.3224614' -2	-8.1489572' -2	-1.0493799' -3	-2.6014367' -3	+1.5887665' -3
+29	+4.3327000' -1	+1.0984240' -1	-6.8811354' -2	-8.0948535' -2	+1.6223968' -3	-1.2588799' -3	+6.8883500' -5
+30	+6.0441000' -1	+9.7012780' -2	-8.0680165' -2	-6.5715872' -2	-1.0272200' -3	-4.5873901' -4	+1.3472267' -4
+31	+7.6454000' -1	+8.3384724' -2	-8.8774799' -2	-3.7377347' -2	-2.1752755' -3	-8.3336556' -5	+1.4204040' -4
+32	+9.2548000' -1	+6.8726087' -2	-9.2729621' -2	-1.2854298' -2	-1.5739130' -3	-1.8002511' -3	-9.8720187' -5
+33	+1.1470000' +0	+4.8000000' -2	-9.3919607' -2	-2.3104086' -5	+0.0000000' +0	-2.6675823' -3	-2.3104086' -5

FIG. 10 PAGE 6

EXAMPLE OF OUTPUT OF SMOOTH

RESULTS OF INTERPOLATION

+0	-1.6399500'	+0	+3.5864425'	-23	+7.6018261'	+0	-1.8490348'	+4	+4.8798554'	+7	-7.2851897'	+10	+4.7972002'	+13
+0	-1.6395467'	+0	+2.0196606'	-3	+3.3694935'	+0	-4.2093559'	+3	+2.3316945'	+7	-5.3503189'	+10	+4.7972002'	+13
+0	-1.6391433'	+0	+3.2365599'	-3	+3.0361041'	+0	+1.3678559'	+3	+5.6393141'	+6	-3.4154482'	+10	+4.7972002'	+13
+0	-1.6387400'	+0	+4.6006568'	-3	+3.7259003'	+0	+1.3888924'	+3	-4.2343376'	+6	-1.4805775'	+10	+4.7972002'	+13
+0	-1.6383367'	+0	+6.1580439'	-3	+3.8326584'	+0	-9.9864155'	+2	-6.3040107'	+6	+4.5429329'	+9	+4.7972002'	+13
+0	-1.6379333'	+0	+7.5629929'	-3	+3.0196885'	+0	-2.6471413'	+3	-5.6970506'	+5	+2.3891640'	+10	+4.7972002'	+13
+0	-1.6375300'	+0	+8.5899996'	-3	+2.2198349'	+0	-4.0900418'	+2	+1.2968572'	+7	+4.3240340'	+10	+4.7972002'	+13
+1	-1.6375300'	+0	+8.5900000'	-3	+2.2198348'	+0	-4.0900205'	+2	+9.6140559'	+4	+1.7884869'	+8	-2.1571742'	+11
+1	-1.6371367'	+0	+9.4326330'	-3	+2.0679964'	+0	-3.5953967'	+2	+1.4980077'	+5	+9.3999838'	+7	-2.1571742'	+11
+1	-1.6367433'	+0	+1.0219829'	-2	+1.9389036'	+0	-2.9553447'	+2	+1.7008710'	+5	+9.1509882'	+6	-2.1571742'	+11
+1	-1.6363500'	+0	+1.0961320'	-2	+1.8356949'	+0	-2.3011351'	+2	+1.5699955'	+5	-7.5697862'	+7	-2.1571742'	+11
+1	-1.6359567'	+0	+1.1667059'	-2	+1.7563455'	+0	-1.7640384'	+2	+1.1053811'	+5	-1.6054671'	+8	-2.1571742'	+11
+1	-1.6355633'	+0	+1.2345187'	-2	+1.6936673'	+0	-1.4753254'	+2	+3.0702800'	+4	-2.4539556'	+8	-2.1571742'	+11
+1	-1.6351700'	+0	+1.3000000'	-2	+1.6353089'	+0	-1.5662665'	+2	-8.2506341'	+4	-3.3024438'	+8	-2.1571742'	+11
+2	-1.6351700'	+0	+1.3000000'	-2	+1.6353089'	+0	-1.5662666'	+2	+4.9375922'	+5	-1.0492658'	+9	+7.9058380'	+11
+2	-1.6347283'	+0	+1.3712522'	-2	+1.6004776'	+0	-2.9537311'	+1	+1.0744285'	+5	-7.0009131'	+8	+7.9058380'	+11
+2	-1.6342867'	+0	+1.4417062'	-2	+1.5891120'	+0	-3.9014382'	+1	-1.2465477'	+5	-3.5091680'	+8	+7.9058380'	+11
+2	-1.6338450'	+0	+1.5112879'	-2	+1.5559371'	+0	-1.1694460'	+2	-2.0253365'	+5	-1.7422864'	+6	+7.9058380'	+11
+2	-1.6334033'	+0	+1.5785878'	-2	+1.4857609'	+0	-1.9521468'	+2	-1.2619379'	+5	+3.4743222'	+8	+7.9058380'	+11
+2	-1.6329617'	+0	+1.6421899'	-2	+1.3934752'	+0	-2.0571135'	+2	+1.0436481'	+5	+6.9660674'	+8	+7.9058380'	+11
+2	-1.6325200'	+0	+1.7020000'	-2	+1.3240548'	+0	-8.0321416'	+1	+4.8914199'	+5	+1.0457811'	+9	+7.9058380'	+11
+3	-1.6325200'	+0	+1.7020000'	-2	+1.3240548'	+0	-8.0321336'	+1	-1.8083363'	+5	+3.7313481'	+8	-2.4027471'	+11
+3	-1.6319950'	+0	+1.7700800'	-2	+1.2652033'	+0	-1.2963110'	+2	-1.8050716'	+4	+2.4699059'	+8	-2.4027471'	+11
+3	-1.6314700'	+0	+1.8347433'	-2	+1.1998555'	+0	-1.1086409'	+2	+7.8506485'	+4	+1.2084636'	+8	-2.4027471'	+11
+3	-1.6309450'	+0	+1.8964275'	-2	+1.1546250'	+0	-5.8788795'	+1	+1.0883797'	+5	-5.2978596'	+6	-2.4027471'	+11
+3	-1.6304200'	+0	+1.9564879'	-2	+1.1378718'	+0	-8.1737235'	+0	+7.2943732'	+4	-1.3144208'	+8	-2.4027471'	+11
+3	-1.6298950'	+0	+2.0162399'	-2	+1.1397025'	+0	+6.2126233'	+0	-2.9176221'	+4	-2.5758631'	+8	-2.4027471'	+11
+3	-1.6293700'	+0	+2.0760000'	-2	+1.1319705'	+0	-5.0398224'	+1	-1.9752183'	+5	-3.8373049'	+8	-2.4027471'	+11
+4	-1.6293700'	+0	+2.0760000'	-2	+1.1319705'	+0	-5.0398256'	+1	+1.7146557'	+4	-1.8761616'	+7	+9.8764090'	+9
+4	-1.6287667'	+0	+2.1434313'	-2	+1.1040521'	+0	-4.3106375'	+1	+7.6246101'	+3	-1.2802850'	+7	+9.8764090'	+9
+4	-1.6281633'	+0	+2.2092794'	-2	+1.0790182'	+0	-4.0474874'	+1	+1.6977854'	+3	-6.8440830'	+6	+9.8764090'	+9
+4	-1.6275600'	+0	+2.2736466'	-2	+1.0547114'	+0	-4.0334695'	+1	-6.3391675'	+2	-8.8531625'	+5	+9.8764090'	+9
+4	-1.6269567'	+0	+2.3365446'	-2	+1.0302829'	+0	-4.0516781'	+1	+6.2950375'	+2	+5.0734505'	+6	+9.8764090'	+9
+4	-1.6263533'	+0	+2.3979733'	-2	+1.0061926'	+0	-3.8852075'	+1	+5.4880468'	+3	+1.1032217'	+7	+9.8764090'	+9
+4	-1.6257500'	+0	+2.4580000'	-2	+9.8420901'	-1	-3.3171521'	+1	+1.3941710'	+4	+1.6990982'	+7	+9.8764090'	+9
+5	-1.6257500'	+0	+2.4580000'	-2	+9.8420901'	-1	-3.3171519'	+1	+2.4257473'	+3	-1.7046424'	+4	-1.8534450'	+4
+5	-1.6250450'	+0	+2.5265765'	-2	+9.6142492'	-1	-3.1465605'	+1	+2.4137249'	+3	-1.7059491'	+4	-1.8534450'	+4
+5	-1.6243400'	+0	+2.5935891'	-2	+9.3984051'	-1	-2.9768169'	+1	+2.4016934'	+3	-1.7072558'	+4	-1.8534450'	+4
+5	-1.6236350'	+0	+2.6591221'	-2	+9.1944981'	-1	-2.8079219'	+1	+2.3896526'	+3	-1.7085625'	+4	-1.8534450'	+4
+5	-1.6229300'	+0	+2.7232594'	-2	+9.0024682'	-1	-2.6398761'	+1	+2.3776026'	+3	-1.7098691'	+4	-1.8534450'	+4
+5	-1.6222250'	+0	+2.7860847'	-2	+8.8222556'	-1	-2.4726801'	+1	+2.3655435'	+3	-1.7111758'	+4	-1.8534450'	+4
+5	-1.6215200'	+0	+2.8476809'	-2	+8.6538003'	-1	-2.3063347'	+1	+2.3534751'	+3	-1.7124825'	+4	-1.8534450'	+4
+6	-1.6215200'	+0	+2.8476809'	-2	+8.6538003'	-1	-2.3063347'	+1	+1.9891874'	+3	-2.6068284'	+5	-4.0404453'	+5
+6	-1.6206983'	+0	+2.9180256'	-2	+8.4707702'	-1	-2.1516933'	+1	+1.7748566'	+3	-2.6101483'	+5	-4.0404453'	+5
+6	-1.6198767'	+0	+2.9869167'	-2	+8.2997227'	-1	-2.0146741'	+1	+1.5602530'	+3	-2.6134682'	+5	-4.0404453'	+5
+6	-1.6190550'	+0	+3.0544466'	-2	+8.1392089'	-1	-1.8952992'	+1	+1.3453766'	+3	-2.6167881'	+5	-4.0404453'	+5
+6	-1.6182333'	+0	+3.1206959'	-2	+7.9877780'	-1	-1.7935913'	+1	+1.1302275'	+3	-2.6201080'	+5	-4.0404453'	+5
+6	-1.6174117'	+0	+3.1857333'	-2	+7.8439775'	-1	-1.7095727'	+1	+9.1480557'	+2	-2.6234279'	+5	-4.0404453'	+5
+6	-1.6165900'	+0	+3.2496155'	-2	+7.7063531'	-1	-1.6432657'	+1	+6.9911090'	+2	-2.6267478'	+5	-4.0404453'	+5
+7	-1.6165900'	+0	+3.2496155'	-2	+7.7063531'	-1	-1.6432657'	+1	+1.1221294'	+3	-1.3369713'	+5	-2.1072086'	+5
+7	-1.6156267'	+0	+3.3231071'	-2	+7.5530593'	-1	-1.5413740'	+1	+9.9323674'	+2	-1.3390012'	+5	-2.1072086'	+5
+7	-1.6146633'	+0	+3.3951674'	-2	+7.4089827'	-1	-1.4519084'	+1	+8.6414851'	+2	-1.3410312'	+5	-2.1072086'	+5
+7	-1.6137000'	+0	+3.4658793'	-2	+7.2729253'	-1	-1.3748877'	+1	+7.3486474'	+2	-1.3430611'	+5	-2.1072086'	+5
+7	-1.6127367'	+0	+3.5353143'	-2	+7.1436874'	-1	-1.3103307'	+1	+6.0538541'	+2	-1.3450911'	+5	-2.1072086'	+5
+7	-1.6117733'	+0	+3.6035324'	-2	+7.0200674'	-1	-1.2582564'	+1	+4.7571053'	+2	-1.3471210'	+5	-2.1072086'	+5
+7	-1.6108100'	+0	+3.6705818'	-2	+6.9008620'	-1	-1.2186835'	+1	+3.4584012'	+2	-1.3491509'	+5	-2.1072086'	+5
+8	-1.6108100'	+0	+3.6705818'	-2	+6.9008619'	-1	-1.2186835'	+1	+4.6555139'	+2	-4.6546281'	+3	-6.9601819'	+3
+8	-1.6096767'	+0	+3.7480201'	-2	+6.7657231'	-1	-1.1662201'	+1	+4.6027168'	+2	-4.6625163'	+3	-6.9601819'	+3
+8	-1.6085433'	+0	+3.8239605'	-2	+6.6364961'	-1	-1.1143555'	+1	+4.5498302'	+2	-4.6704045'	+3	-6.9601819'	+3
I	X		$\hat{g}$		$\hat{g}^{I}$		$\hat{g}^{II}$		$\hat{g}^{III}$		$\hat{g}^{IV}$		$\hat{g}^{V}$	

FIG. 10 PAGE 7

EXAMPLE OF OUTPUT OF SMOOTH

+8	-1.6074100'	+0	+3.8984695'	-2	+6.5131131'	-1	-1.0630909'	+1	+4.4968543'	+2	-4.6782927'	+3	-6.9601819'	+3
+8	-1.6062767'	+0	+3.9716129'	-2	+6.3955061'	-1	-1.0124272'	+1	+4.4437889'	+2	-4.6861809'	+3	-6.9601819'	+3
+8	-1.6051433'	+0	+4.0434558'	-2	+6.2836069'	-1	-9.6236537'	+0	+4.3906342'	+2	-4.6940692'	+3	-6.9601819'	+3
+8	-1.6040100'	+0	+4.1140626'	-2	+6.1773472'	-1	-9.1290649'	+0	+4.3373900'	+2	-4.7019574'	+3	-6.9601819'	+3
+9	-1.6040100'	+0	+4.1140626'	-2	+6.1773472'	-1	-9.1290648'	+0	+3.0048038'	+2	-5.7076241'	+3	-9.1110337'	+3
+9	-1.6026700'	+0	+4.1960314'	-2	+6.0576925'	-1	-8.7315491'	+0	+2.9282399'	+2	-5.7198329'	+3	-9.1110337'	+3
+9	-1.6013300'	+0	+4.2764323'	-2	+5.9432958'	-1	-8.3443038'	+0	+2.8515123'	+2	-5.7320417'	+3	-9.1110337'	+3
+9	-1.5999900'	+0	+4.3553346'	-2	+5.8340192'	-1	-7.9673511'	+0	+2.7746211'	+2	-5.7442505'	+3	-9.1110337'	+3
+9	-1.5986500'	+0	+4.4328062'	-2	+5.7297247'	-1	-7.6007127'	+0	+2.6975664'	+2	-5.7564593'	+3	-9.1110337'	+3
+9	-1.5973100'	+0	+4.5089129'	-2	+5.6302740'	-1	-7.2444106'	+0	+2.6203480'	+2	-5.7686681'	+3	-9.1110337'	+3
+9	-1.5959700'	+0	+4.5837186'	-2	+5.5355283'	-1	-6.8984668'	+0	+2.5429661'	+2	-5.7808768'	+3	-9.1110337'	+3
+10	-1.5959700'	+0	+4.5837186'	-2	+5.5355283'	-1	-6.8984667'	+0	+2.3771431'	+2	-1.3986362'	+4	-2.6784836'	+4
+10	-1.5943917'	+0	+4.6702436'	-2	+5.4295166'	-1	-6.5407128'	+0	+2.1560581'	+2	-1.4028637'	+4	-2.6784836'	+4
+10	-1.5928133'	+0	+4.7551386'	-2	+5.3288759'	-1	-6.2179062'	+0	+1.9343058'	+2	-1.4070912'	+4	-2.6784836'	+4
+10	-1.5912350'	+0	+4.8384838'	-2	+5.2330536'	-1	-5.9301520'	+0	+1.7118863'	+2	-1.4113188'	+4	-2.6784836'	+4
+10	-1.5896567'	+0	+4.9203511'	-2	+5.1414958'	-1	-5.6775558'	+0	+1.4887995'	+2	-1.4155463'	+4	-2.6784836'	+4
+10	-1.5880783'	+0	+5.0008032'	-2	+5.0536466'	-1	-5.4602228'	+0	+1.2650455'	+2	-1.4197739'	+4	-2.6784836'	+4
+10	-1.5865000'	+0	+5.0798944'	-2	+4.9689487'	-1	-5.2782582'	+0	+1.0406243'	+2	-1.4240014'	+4	-2.6784836'	+4
+11	-1.5865000'	+0	+5.0798944'	-2	+4.9689487'	-1	-5.2782582'	+0	+3.0446859'	+2	-3.3602614'	+4	-8.4530917'	+4
+11	-1.5844750'	+0	+5.1794732'	-2	+4.8678409'	-1	-4.7307222'	+0	+2.3624998'	+2	-3.3773789'	+4	-8.4530917'	+4
+11	-1.5824500'	+0	+5.2771074'	-2	+4.7764196'	-1	-4.3216798'	+0	+1.6768475'	+2	-3.3944964'	+4	-8.4530917'	+4
+11	-1.5804250'	+0	+5.3729646'	-2	+4.6918733'	-1	-4.0518329'	+0	+9.8772882'	+1	-3.4116139'	+4	-8.4530917'	[illegible]
+11	-1.5784000'	+0	+5.4671556'	-2	+4.6113761'	-1	-3.9218836'	+0	+2.9514386'	+1	-3.4287314'	+4	[illegible]	
+11	-1.5763750'	+0	+5.5597335'	-2	+4.5320879'	-1	-3.9325336'	+0	-4.0090740'	+1	[illegible]			
+11	-1.5743500'	+0	+5.6506940'	-2	+4.4511547'	-1	-4.0844850'	+0	[illegible]					
+12	-1.5743500'	+0	+5.6506940'	-2	+4.4511547'	-1	[illegible]							
+12	-1.5721850'	+0	+5.7461169'	-2	[illegible]									
+12	-1.5700200'	+0	+5.8397001'											
+12	-1.5678550'	[illegible]												
+12	[illegible]													

														[illegible]
												[illegible]	-2.1993637'	+0
											[illegible]	+0	-2.1993637'	+0
									[illegible]	-2	-1.6936846'	+0	-2.1993637'	+0
							[illegible]	-2	-4.9279954'	-3	-1.7564178'	+0	-2.1993637'	+0
						-2	-6.5715872'	-2	-5.5921566'	-2	-1.8191510'	+0	-2.1993637'	+0
				-2	-8.0680165'	-2	-6.5715872'	-2	+2.1298483'	-1	-4.1694557'	-1	-6.1542843'	-1
			+9.4836823'	-2	-8.2359495'	-2	-6.0182100'	-2	+2.0163808'	-1	-4.3337033'	-1	-6.1542843'	-1
	[illegible]	-1	+9.2617982'	-2	-8.3895231'	-2	-5.4957003'	-2	+1.8985297'	-1	-4.4979509'	-1	-6.1542843'	-1
+30	+6.8447500'	-1	+9.0359978'	-2	-8.5295767'	-2	-5.0052281'	-2	+1.7762951'	-1	-4.6621985'	-1	-6.1542843'	-1
+30	+7.1116333'	-1	+8.8066304'	-2	-8.6569809'	-2	-4.5479631'	-2	+1.6496771'	-1	-4.8264461'	-1	-6.1542843'	-1
+30	+7.3785167'	-1	+8.5740216'	-2	-8.7726376'	-2	-4.1250754'	-2	+1.5186755'	-1	-4.9906937'	-1	-6.1542843'	-1
+30	+7.6454000'	-1	+8.3384724'	-2	-8.8774799'	-2	-3.7377347'	-2	+1.3832905'	-1	-5.1549412'	-1	-6.1542843'	-1
+31	+7.6454000'	-1	+8.3384724'	-2	-8.8774799'	-2	-3.7377347'	-2	+1.7196189'	-1	-2.2753976'	-1	-2.9601971'	-1
+31	+7.9136333'	-1	+8.0990590'	-2	-8.9716260'	-2	-3.2847565'	-2	+1.6575202'	-1	-2.3547999'	-1	-2.9601971'	-1
+31	+8.1818667'	-1	+7.8572812'	-2	-9.0538477'	-2	-2.8487208'	-2	+1.5932917'	-1	-2.4342023'	-1	-2.9601971'	-1
+31	+8.4501000'	-1	+7.6134528'	-2	-9.1246070'	-2	-2.4301990'	-2	+1.5269334'	-1	-2.5136046'	-1	-2.9601971'	-1
+31	+8.7183333'	-1	+7.3678747'	-2	-9.1843814'	-2	-2.0297624'	-2	+1.4584452'	-1	-2.5930070'	-1	-2.9601971'	-1
+31	+8.9865667'	-1	+7.1208351'	-2	-9.2336638'	-2	-1.6479823'	-2	+1.3878272'	-1	-2.6724093'	-1	-2.9601971'	-1
+31	+9.2548000'	-1	+6.8726087'	-2	-9.2729621'	-2	-1.2854298'	-2	+1.3150793'	-1	-2.7518117'	-1	-2.9601971'	-1
+32	+9.2548000'	-1	+6.8726087'	-2	-9.2729621'	-2	-1.2854298'	-2	+8.4894934'	-2	-2.0878882'	-1	-4.7026512'	-1
+32	+9.6240000'	-1	+6.5294444'	-2	-9.3148130'	-2	-9.8662204'	-3	+7.6865945'	-2	-2.2615101'	-1	-4.7026512'	-1
+32	+9.9932000'	-1	+6.1849318'	-2	-9.3461936'	-2	-7.1864057'	-3	+6.8195944'	-2	-2.4351320'	-1	-4.7026512'	-1
+32	+1.0362400'	+0	+5.8394358'	-2	-9.3682859'	-2	-4.8385206'	-3	+5.8884930'	-2	-2.6087539'	-1	-4.7026512'	-1
+32	+1.0731600'	+0	+5.4932763'	-2	-9.3823589'	-2	-2.8462312'	-3	+4.8932905'	-2	-2.7823758'	-1	-4.7026512'	-1
+32	+1.1100800'	+0	+5.1467245'	-2	-9.3897692'	-2	-1.2332036'	-3	+3.8339868'	-2	-2.9559977'	-1	-4.7026512'	-1
+32	+1.1470000'	+0	+4.8000000'	-2	-9.3919607'	-2	-2.3104090'	-5	+2.7105818'	-2	-3.1296195'	-1	-4.7026512'	-1
I	X		$\hat{g}$		$\hat{g}^{I}$		$\hat{g}^{II}$		$\hat{g}^{III}$		$\hat{g}^{IV}$		$\hat{g}^{V}$	

FIG. 10 PAGE 8

EXAMPLE OF OUTPUT OF SMOOTH

CORRECTED AEROFOIL SECTION
I, NEW VALUES. OLD VALUES MINUS NEW VALUES

+0	-1.6399500'	+0	+3.5864425'	-23	+1.4400000'	+0	+4.1021800'	+1	-3.5864425'	-23	+0.0000000'	+0	+0.0000000'	+0
+1	-1.6375300'	+0	+8.5900000'	-3	+1.1475400'	+0	+2.8340000'	+1	+0.0000000'	+0	+0.0000000'	+0	+0.0000000'	+0
+2	-1.6351700'	+0	+1.3000000'	-2	+1.0219600'	+0	+2.2239000'	+1	+0.0000000'	+0	+0.0000000'	+0	+0.0000000'	+0
+3	-1.6325200'	+0	+1.7020000'	-2	+9.2394000'	-1	+1.7583000'	+1	+0.0000000'	+0	+0.0000000'	+0	+0.0000000'	+0
+4	-1.6293700'	+0	+2.0760000'	-2	+8.4722000'	-1	+1.4626000'	+1	+0.0000000'	+0	+0.0000000'	+0	+0.0000000'	+0
+5	-1.6257500'	+0	+2.4580000'	-2	+7.7744000'	-1	+1.2009000'	+1	+0.0000000'	+0	+0.0000000'	+0	+0.0000000'	+0
+6	-1.6215200'	+0	+2.8476809'	-2	+7.1335548'	-1	+9.9719788'	+0	-6.8087339'	-6	+1.4345244'	-3	-4.1778829'	-2
+7	-1.6165900'	+0	+3.2496155'	-2	+6.5657743'	-1	+8.1662912'	+0	+9.3844942'	-5	-2.8743443'	-4	+5.6087714'	-3
+8	-1.6108100'	+0	+3.6705818'	-2	+6.0404137'	-1	+6.7946133'	+0	-5.8178070'	-6	-1.6136932'	-4	+5.7866539'	-3
+9	-1.6040100'	+0	+4.1140626'	-2	+5.5335778'	-1	+5.6215309'	+0	-3.0626171'	-5	+4.8222264'	-4	-6.8309077'	-3
+10	-1.5959700'	+0	+4.5837186'	-2	+5.0556682'	-1	+4.6198511'	+0	+2.2814101'	-5	+7.1318402'	-4	-6.6510977'	-3
+11	-1.5865000'	+0	+5.0798944'	-2	+4.6116042'	-1	+3.7908870'	+0	+1.5105558'	-4	+5.4957992'	-4	-3.0869986'	-3
+12	-1.5743500'	+0	+5.6506940'	-2	+4.1878453'	-1	+3.1144589'	+0	-2.0694048'	-4	+1.2554713'	-3	-1.8588909'	-3
+13	-1.5613600'	+0	+6.1970965'	-2	+3.7890890'	-1	+2.5357491'	+0	-1.2096500'	-4	+1.3510987'	-3	-6.9491092'	-3
+14	-1.5468100'	+0	+6.7456729'	-2	+3.4320948'	-1	+2.1441363'	+0	-4.0672922'	-4	+1.0505194'	-3	-2.3362681'	-3
+15	-1.5301500'	+0	+7.3079595'	-2	+3.0876756'	-1	+1.8138015'	+0	+5.0404646'	-5	+3.2244059'	-4	-6.0150876'	-4
+16	-1.5137000'	+0	+7.8067264'	-2	+2.8055680'	-1	+1.6315550'	+0	-1.0972644'	-3	-1.7679617'	-4	+3.3450322'	-3
+17	-1.4967600'	+0	+8.2690959'	-2	+2.5258496'	-1	+1.5524003'	+0	-1.0909586'	-3	+4.4503816'	-4	-3.7002609'	-3
+18	-1.4800900'	+0	+8.6767710'	-2	+2.2701383'	-1	+1.5980056'	+0	+6.4228958'	-4	+5.4617075'	-4	-6.0564569'	-4
+19	-1.4704400'	+0	+8.8904887'	-2	+2.0825611'	-1	+1.9826023'	+0	+1.1051134'	-3	-4.7610595'	-4	+5.9773711'	-4
+20	-1.4622700'	+0	+9.0558387'	-2	+1.8988084'	-1	+2.8913487'	+0	+1.9161326'	-4	+1.4916297'	-4	-2.4867544'	-4
+21	-1.4574500'	+0	+9.1450527'	-2	+1.7632744'	-1	+2.7309925'	+0	-8.5052660'	-4	-2.6174422'	-3	+3.8075398'	-3
+22	-1.3942200'	+0	+1.0019759'	-1	+1.2375828'	-1	+4.0963132'	-1	-8.4275880'	-3	+2.3517154'	-3	-3.5132054'	-4
+23	-1.2858800'	+0	+1.1173546'	-1	+9.1414210'	-2	+2.4916324'	-1	-1.5454641'	-3	-5.2420983'	-4	+5.6764874'	-5
+24	-1.1392400'	+0	+1.2298127'	-1	+6.3659946'	-2	+1.6910133'	-1	+4.0587271'	-3	+2.6005416'	-4	+1.0866984'	-4
+25	-9.4031000'	-1	+1.3283063'	-1	+3.7695622'	-2	+9.6629109'	-2	+1.0193726'	-3	-1.9356222'	-3	+5.0908910'	-3
+26	-5.3171000'	-1	+1.4085696'	-1	+3.2701216'	-3	+7.1925579'	-2	-4.0696209'	-4	+2.5987839'	-4	-5.4185793'	-3
+27	-5.6950000'	-2	+1.3427420'	-1	-3.1061038'	-2	+7.2708189'	-2	-2.7420478'	-4	+2.8710377'	-3	-6.1918900'	-4
+28	+2.3045000'	-1	+1.2222062'	-1	-5.3174440'	-2	+8.1144521'	-2	+1.0493799'	-3	+2.5944404'	-3	+1.6154791'	-3
+29	+4.3327000'	-1	+1.0984240'	-1	-6.8703055'	-2	+8.0376982'	-2	-1.6223968'	-3	+1.2530545'	-3	+8.9018483'	-5
+30	+6.0441000'	-1	+9.7012780'	-2	-8.0505789'	-2	+6.5079408'	-2	+1.0272200'	-3	+4.5578898'	-4	+1.4059205'	-4
+31	+7.6454000'	-1	+8.3384724'	-2	-8.8542686'	-2	+3.6939806'	-2	+2.1752755'	-3	+8.2685526'	-5	+1.4119384'	-4
+32	+9.2548000'	-1	+6.8726087'	-2	-9.2465197'	-2	+1.2690266'	-2	+1.5739130'	-3	+1.7851967'	-3	-9.1265751'	-5
+33	+1.1470000'	+0	+4.8000000'	-2	-9.3644908'	-2	+2.2801725'	-5	+0.0000000'	+0	+2.6449083'	-3	-2.2801725'	-5
I	X_i		$\hat{Y}_i$		$\hat{\theta}_i$		$(1/\hat{R})_i$		$Y_i - \hat{Y}_i$		$\theta_i - \hat{\theta}_i$		$(1/R)_i - (1/\hat{R})_i$	

FIG. 10 PAGE 9

EXAMPLE OF OUTPUT OF SMOOTH

```
≠BEGIN≠ ≠COMMENT≠COMPUTATION OF THE SERIES LAMBDA(/N/),D/DN LAMBDA(/N/),
    LAMBDA(/N+1/2/),LAMBDA(/-N-1/2/),L(/N/) AND L(/-N/)
    FOR  N=0(1)M,J=1,2,3,I=1,2.,

   ≠REAL≠ TOL,EPSZERO,ALFA,GAMMA., ≠INTEGER≠ CASE,M.,
   ≠ARRAY≠ PARAM(/1..6/).,

   INARRAY(40,PARAM).,
   M.=PARAM(/1/)., TOL.=PARAM(/2/)., CASE.=PARAM(/3/).,
   EPSZERO.=PARAM(/4/)., ALFA.=PARAM(/5/)., GAMMA.=PARAM(/6/).,

 ≠BEGIN≠ ≠REAL≠ PI,ABSZ1,ABSZ2,TOLF,TOLZ1,
     TOLZ2,TOLEZ1,TOLEZ2,FACT,H,SUM,LN2.,
   ≠INTEGER≠ I,K,P,N,J.,
   ≠BOOLEAN≠ CONVS1,CONVS2,CONVS3.,
   ≠ARRAY≠ Z1,Z2,Z1DZ2,EPS,EZ1,EZ2,F,S1,S2,Z,EZ,T1,T2,EN,L,LM2,LP2,LM3,
     S3,T3,ZZ,IZ12,T,R,DF,DNL,D,DS1,DS2,DT1,DT2,LNZ1,
        LP3(/1..2/),FGAM(/1..2,1..2/),Z2P,EZ2P(/1..2,-M-1..0/),
     DB,B,Z1P,EZ1P,EP(/1..2,0..M+1/),LARRAY(/1..2,1..11,0..M,1..2/).,

 ≠PROCEDURE≠ CONSTANTS.,
   ≠BEGIN≠
   ≠REAL≠ H,CZ1,CZ2,MOD,ARC.,
   ≠ARRAY≠ EPS1,H1,H2(/1..2/).,

   ≠PROCEDURE≠ CMD(A,B,Z,T)., ≠VALUE≠ T., ≠REAL≠ T., ≠ARRAY≠ A,B,Z.,
   ≠BEGIN≠≠REAL≠ A1,A2,B1,B2,MOD.,
     A1.=A(/1/)., A2.=A(/2/)., B1.=B(/1/)., B2.=B(/2/).,
     MOD.=≠IF≠ 0 ≠LESS≠ T ≠THEN≠ 1.0 ≠ELSE≠ 1.0/(B1*B1+B2*B2).,
     Z(/1/).=MOD*(A1*B1-T*A2*B2).,
     Z(/2/).=MOD*(A2*B1+T*A1*B2).,
   ≠END≠ CMD.,

   PI.=4.0*ARCTAN(1.0).,
   EPS(/1/).=EPSZERO*COS(2.0*ALFA).,
   EPS1(/1/).=1.0+EPS(/1/).,
   EPS(/2/).=EPS1(/2/).=EPSZERO*SIN(2.0*ALFA).,
   CMD(EPS,EPS1,H1,-1).,
   CMD(H1,EPS1,H2,-1).,
   H.=4.0-(0.5*GAMMA/PI) ≠POWER≠ 2.,
   CZ1.=1.0-H*H2(/1/)., CZ2.=-H*H2(/2/).,
   MOD.=(CZ1*CZ1+CZ2*CZ2) ≠POWER≠ 0.25.,
   ARC.=0.5*(ARCTAN(CZ2/CZ1)+(≠IF≠ CZ1 ≠LESS≠ 0 ≠THEN≠ SIGN(CZ2)*PI
     ≠ELSE≠ 0.0)).,
   H2(/1/).=0.5*(1.0-MOD*COS(ARC)).,
   H2(/2/).=-0.5*MOD*SIN(ARC).,
   CMD(H2,H1,Z1,-1).,
   H2(/1/).=1.0-H2(/1/).,
   H2(/2/).=-H2(/2/).,
   CMD(H2,H1,Z2,-1).,
   ABSZ1.=SQRT(Z1(/1/) ≠POWER≠ 2+Z1(/2/) ≠POWER≠ 2).,
   ABSZ2.=SQRT(Z2(/1/) ≠POWER≠ 2+Z2(/2/) ≠POWER≠ 2).,
   CMD(Z1,Z2,Z1DZ2,-1).,
   ≠END≠ PROCEDURE CONSTANTS.,

   ≠PROCEDURE≠ COMUDI1(A,B,C,T)., ≠VALUE≠ T.,
     ≠INTEGER≠ T., ≠ARRAY≠ A,B,C.,
   ≠BEGIN≠ ≠REAL≠ MOD,P1,P2,Q1,Q2.,
```

APPENDIX A
LISTING OF COEFF

```
      P1.=A(/1/)., P2.=A(/2/)., Q1.=B(/1/)., Q2.=B(/2/).,
      MOD.=≠IF≠ T ≠LESS≠ 0 ≠THEN≠ 1.0/(Q1*Q1+Q2*Q2) ≠ELSE≠ 1.0.,
      C(/1/).=(P1*Q1-T*P2*Q2)*MOD.,
      C(/2/).=(T*P1*Q2+P2*Q1)*MOD.,
    ≠END≠ COMUDI1.,

    ≠PROCEDURE≠ COMUDI2(A,J,B,K,C,T)., ≠VALUE≠ J,K,T.,
      ≠INTEGER≠ J,K,T., ≠ARRAY≠ A,B,C.,
    ≠BEGIN≠ ≠REAL≠ MOD,P1,P2,Q1,Q2.,
      P1.=A(/1,J/)., P2.=A(/2,J/)., Q1.=B(/1,K/)., Q2.=B(/2,K/).,
      MOD.=≠IF≠ T ≠LESS≠ 0 ≠THEN≠ 1.0/(Q1*Q1+Q2*Q2) ≠ELSE≠ 1.0.,
      C(/1/).=(P1*Q1-T*P2*Q2)*MOD.,
      C(/2/).=(T*P1*Q2+P2*Q1)*MOD.,
    ≠END≠ COMUDI2.,

  ≠PROCEDURE≠ COHYPF(A,B,C)., ≠VALUE≠ A,B,C.,
    ≠REAL≠ A,B,C.,
  ≠BEGIN≠ ≠REAL≠ PART,T1,T2,S1,S2,H., ≠INTEGER≠ K,KM.,
    S1.=T1.=1., S2.=T2.=0., KM.=0.,
  TERM.. K.=KM+1.,
    PART.=(A+KM)*(B+KM)/(C+KM)/K.,
    H.=T1*Z1DZ2(/1/)-T2*Z1DZ2(/2/).,
    T2.=(T1*Z1DZ2(/2/)+T2*Z1DZ2(/1/))*PART.,
    T1.=PART*H.,
    S1.=S1+T1., S2.=S2+T2., KM.=K.,
    ≠IF≠ ABS(T1/S1) ≠GREATER≠ TOLF ≠OR≠ ABS(T2/S2) ≠GREATER≠ TOLF
    ≠THEN≠ ≠GOTO≠ TERM.,
    F(/1/).=FACT*S1., F(/2/).=FACT*S2
  ≠END≠ COHYPF.,

  ≠PROCEDURE≠ DCOHYPF(A,DA,B,DB,C,DC).,
  ≠VALUE≠ A,DA,B,DB,C,DC., ≠REAL≠ A,DA,B,DB,C,DC.,
  ≠BEGIN≠ ≠REAL≠ DTERM., ≠INTEGER≠ K., ≠ARRAY≠ KLAD,TERM,S,T,DT(/1..2/).,
    S(/1/).=S(/2/).=DT(/1/).=DT(/2/).=T(/2/).=TERM(/2/).=0.,
    T(/1/).=1., K.=-1.,
  AA.. K.=K+1., TERM(/1/).=(A+K)*(B+K)/(C+K)/(K+1).,
    DTERM.=(((B+K)*DA+(A+K)*DB)/(K+1)-TERM(/1/)*DC)/(C+K).,
    KLAD(/1/).=DTERM*T(/1/)+TERM(/1/)*DT(/1/).,
    KLAD(/2/).=DTERM*T(/2/)+TERM(/1/)*DT(/2/).,
    COMUDI1(KLAD,Z1DZ2,DT,1).,
    S(/1/).=S(/1/)+DT(/1/)., S(/2/).=S(/2/)+DT(/2/).,
    ≠IF≠ ABS(DT(/1/)/S(/1/))≠GREATER≠ TOLF ≠OR≠ ABS(DT(/2/)/S(/2/))
    ≠GREATER≠ TOLF ≠THEN≠
    ≠BEGIN≠ COMUDI1(TERM,Z1DZ2,KLAD,1).,
      COMUDI1(KLAD,T,T,1)., ≠GOTO≠ AA
    ≠END≠.,
    DF(/1/).=FACT*S(/1/)., DF(/2/).=FACT*S(/2/)
  ≠END≠ DCOHYPF.,

  ≠PROCEDURE≠ TEST(SUM,TERM,TOL,REPEAT,READY).,
    ≠REAL≠ TOL., ≠BOOLEAN≠ READY., ≠ARRAY≠ SUM,TERM., ≠LABEL≠ REPEAT.,
  ≠BEGIN≠ SUM(/1/).=SUM(/1/)+TERM(/1/)., SUM(/2/).=SUM(/2/)+TERM(/2/).,
    ≠IF≠ (TERM(/1/)*TERM(/1/)+TERM(/2/)*TERM(/2/))/
     (SUM(/1/)*SUM(/1/)+SUM(/2/)*SUM(/2/)) ≠GREATER≠ TOL*TOL
    ≠THEN≠ ≠GOTO≠ REPEAT ≠ELSE≠
    ≠BEGIN≠ READY.=≠TRUE≠., OUTPUT(41,≠(≠//4ZD≠)≠,P) ≠END≠
  ≠END≠ TEST.,
```

```
≠PROCEDURE≠ POWERS(Z,S,ZP)., ≠VALUE≠ S., ≠INTEGER≠ S., ≠ARRAY≠ Z,ZP.,
≠BEGIN≠ ≠REAL≠ MOD,ARC,PARC,MODP., ≠INTEGER≠ P,MS.,
  MOD.=(Z(/1/)*Z(/1/)+Z(/2/)*Z(/2/))≠POWER≠(.5*S).,
  MS.=(M+1)*S.,  MODP.=1.,
  ARC.=ARCTAN(Z(/2/)/Z(/1/)).,  ≠IF≠ Z(/1/) ≠LESS≠ 0 ≠THEN≠ ARC.=ARC+PI.,
  ≠FOR≠ P.=0 ≠STEP≠ S ≠UNTIL≠ MS ≠DO≠
  ≠BEGIN≠ PARC.=P*ARC.,
    ZP(/1,P/).=MODP*COS(PARC).,
    ZP(/2,P/).=MODP*SIN(PARC) .,
    MODP.=MODP*MOD
  ≠END≠.,
≠END≠ POWERS.,


  CONSTANTS.,
  OUTPUT(61,≠(≠2(/,7(19S))≠)≠,≠(≠CASE≠)≠,≠(≠EPSZERO≠)≠,≠(≠ALFA≠)≠,
    ≠(≠GAMMA≠)≠,≠(≠RE ZETA1≠)≠,≠(≠IM ZETA1≠)≠,≠(≠RE ZETA2≠)≠,
    ≠(≠IM ZETA2≠)≠,≠(≠ABSZETA1≠)≠,≠(≠ABSZETA2≠)≠,≠(≠RE Z1 DIV Z2≠)≠,
    ≠(≠IM Z1 DIV Z2≠)≠,≠(≠RE EPS≠)≠,≠(≠IM EPS≠)≠).,
  OUTPUT(61,≠(≠2(/,7(+D.10D≠+ZD2B))≠)≠,
    CASE,EPSZERO,ALFA,GAMMA,Z1(/1/),Z1(/2/),Z2(/1/),
   Z2(/2/),ABSZ1,ABSZ2,Z1DZ2(/1/),Z1DZ2(/2/),EPS(/1/),
    EPS(/2/)).,

  OUTPUT(41,≠(≠///,≠(≠TOLERANCE ≠)≠,D≠+ZD,//,
≠(≠HIGHEST SUBSCRIPT OF COEFFICIENTS≠)≠,3ZD≠)≠,TOL,M).,
OUTPUT(41,≠(≠//,≠(≠NUMBERS OF TERMS IN POWER SERIES≠)≠≠)≠).,

  GAMMA.=.5*GAMMA/PI.,
  FGAM(/1,1/).=.5*GAMMA/SQRT(1-.25*GAMMA*GAMMA).,
  FGAM(/1,2/).=PI/FGAM(/1,1/)., FGAM(/1,1/).=PI*FGAM(/1,1/).,
  TOLF.=.1*TOL.,
  FGAM(/2,1/).=FGAM(/2,2/).=0.,
  COMUDI1(EPS,Z1,EZ1,1)., COMUDI1(EPS,Z2,EZ2,1).,
  TOLZ1.=(1.0/ABSZ1-1.0)*TOL.,
  TOLZ2.=(ABSZ2-1.0)*TOL.,
  TOLEZ1.=(1.0/ABSZ1/EPSZERO-1.0)*TOL.,
  TOLEZ2.=(EPSZERO*ABSZ2-1.0)*TOL.,
  POWERS(Z1,1,Z1P)., POWERS(EZ1,1,EZ1P)., POWERS(EPS,1,FP).,
  POWERS(Z2,-1,Z2P)., POWERS(EZ2,-1,EZ2P).,
  ≠FOR≠ I.=1,2 ≠DO≠
  ≠BEGIN≠ K.=I-1.,
    ≠FOR≠ P.=1 ≠STEP≠ 1 ≠UNTIL≠ M+K ≠DO≠
    ≠BEGIN≠ FACT.=≠IF≠ P=1 ≠THEN≠ (8*I-10)/3 ≠ELSE≠
      (P-1)/(P+.5-K)*FACT.,
      COHYPF(K-.5,P,P+1.5-K).,
     B(/1,P/).=F(/1/)., B(/2,P/).=F(/2/)
    ≠END≠.,
    P.=M+K., S1(/1/).=S1(/2/).=S2(/1/).=S2(/2/).=0.,
    CONVS1.=CONVS2.=≠FALSE≠.,
    Z(/1/).=Z1P(/1,P/)., Z(/2/).=Z1P(/2,P/).,
    EZ(/1/).=EZ1P(/1,P/)., EZ(/2/).=EZ1P(/2,P/).,
  B5ORB6..
    P.=P+1.,
    FACT.=(P-1)/(P+.5-K)*FACT.,
    COHYPF(K-.5,P,P+1.5-K).,
    ≠IF≠ CONVS2 ≠THEN≠ ≠GOTO≠ BERS1.,
    COMUDI1(EZ1,EZ,EZ,1)., COMUDI1(EZ,F,T2,1).,
```

```
          TEST(S2,T2,TOLEZ1,BERS1,CONVS2).,
      BERS1..  COMUDI1(Z1,Z,Z,1)., COMUDI1(Z,F,T1,1).,
          TEST(S1,T1,TOLZ1,B5ORB6,CONVS1).,
        ≠COMMENT≠ RECURSIEFORMULES..  .,
        ≠FOR≠ N.=M ≠STEP≠ -1 ≠UNTIL≠ 1-K ≠DO≠
        ≠BEGIN≠
          COMUDI2(B,N+K,Z1P,N+K,T1,1).,
          COMUDI2(B,N+K,EZ1P,N+K,T2,1).,
          S1(/1/).=S1(/1/)+T1(/1/)., S1(/2/).=S1(/2/)+T1(/2/).,
          S2(/1/).=S2(/1/)+T2(/1/)., S2(/2/).=S2(/2/)+T2(/2/).,
          ≠IF≠ N=0 ≠THEN≠ ≠GOTO≠ AA.,
          EN(/1/).=EP(/1,N/)., EN(/2/).=EP(/2,N/).,
          COMUDI1(S2,EN,L,-1).,
          H.=K*(1/N+1)-1., ≠COMMENT≠ RESP. -1 EN 1/N.,
          COMUDI2(EP,N,FGAM,I,LP3,1).,
          ≠FOR≠ J.=1,2 ≠DO≠
          ≠BEGIN≠ LM2(/J/).=H*(S1(/J/)+L(/J/)).,
            LM3(/J/).=LM2(/J/)-H*FGAM(/J,I/).,
            LP3(/J/).=(1-2*K)*H*LP3(/J/).,
            LP2(/J/).=LP3(/J/)+(1-2*K)*H*FGAM(/J,I/).,
            LARRAY(/I,6,N,J/).=LP2(/J/).,
            LARRAY(/I,7,N,J/).=LM2(/J/).,
            LARRAY(/I,10,N,J/).=LP3(/J/).,
            LARRAY(/I,11,N,J/).=LM3(/J/).,
          ≠END≠.,
        ≠END≠.,
        AA..
        ≠FOR≠ N.=1 ≠STEP≠ 1 ≠UNTIL≠ M ≠DO≠
        ≠BEGIN≠ COMUDI1(S2,EPS,S2,1).,
          ≠FOR≠ J.=1,2 ≠DO≠
          LARRAY(/I,3,N,J/).=(S1(/J/)+S2(/J/))/(≠IF≠ I=1 ≠THEN≠ 1 ≠ELSE≠ N).,
        ≠END≠.,
      ≠END≠ I-CYCLE.,
      ≠COMMENT≠ COMPUTATION OF THE SERIES LAMBDA.,
          FGAM(/1,1/).=FGAM(/1,2/).=0.,
          FGAM(/2,1/).=.5*GAMMA/SQRT(1-.25*GAMMA*GAMMA).,
          FGAM(/2,2/).=1.0/FGAM(/2,1/).,
      ≠FOR≠ I.=1,2 ≠DO≠
      ≠BEGIN≠ K.=I-1.,
          H.=ARCTAN(Z1(/2/)/Z1(/1/))*.5.,
          IZ12(/1/).=ABSZ1≠POWER≠(K-.5)*(1-2*K)*SIN(H).,
          IZ12(/2/).=ABSZ1≠POWER≠(K-.5)*COS(H).,
          ≠FOR≠ P.=0 ≠STEP≠ 1 ≠UNTIL≠ M+1 ≠DO≠
          ≠BEGIN≠ FACT.=≠IF≠ P=0 ≠THEN≠1 ≠ELSE≠ (P+K-1.5)/P*FACT.,
            COHYPF(K-.5,P+K-.5,P+1).,
            B(/1,P/).=F(/1/)., B(/2,P/).=F(/2/)
          ≠END≠.,
          P.=M+1., S1(/1/).=S1(/2/).=S2(/1/).=S2(/2/).=S3(/1/).=S3(/2/).=0.,
          CONVS1.=CONVS2.=CONVS3.=≠FALSE≠.,
          EZ(/1/).=EZ1P(/1,P/)., EZ(/2/).=EZ1P(/2,P/).,
          ZZ(/1/).=Z2P(/1,-P/)., ZZ(/2/).=Z2P(/2,-P/).,
          Z(/1/).=Z1P(/1,P/)., Z(/2/).=Z1P(/2,P/).,
          B2ORB4..
          P.=P+1.,
          FACT.=(P+K-1.5)/P*FACT.,
          COHYPF(K-.5,P+K-.5,P+1).,
          ≠IF≠ CONVS1 ≠THEN≠ ≠GOTO≠ BERS2.,
          COMUDI1(EZ1,EZ,EZ,1)., COMUDI1(EZ,F,T1,1).,
```

```
          TEST(S1,T1,TOLEZ1,BERS2,CONVS1).,                              COEFF
      BERS2..  ≠IF≠ CONVS2 ≠THEN≠ ≠GOTO≠ BERS3.,                         COEFF
          COMUDI1(ZZ,Z2,ZZ,-1)., COMUDI1(ZZ,F,T2,1).,                    COEFF
          TEST(S2,T2,TOLZ2,BERS3,CONVS2).,                               COEFF
      BERS3..  COMUDI1(Z1,Z,Z,1)., COMUDI1(Z,F,T3,1).,                   COEFF
          TEST(S3,T3,TOLZ1,B2ORB4,CONVS3).,                              COEFF
          ≠IF≠ ≠NOT≠ CONVS1 ≠OR≠ ≠NOT≠  CONVS2 ≠THEN≠ ≠GOTO≠ B2ORB4.,    COEFF
      ≠COMMENT≠ RECURSIEFORMULES.. .,                                    COEFF
          ≠FOR≠ N.=M ≠STEP≠ -1 ≠UNTIL≠ 0 ≠DO≠                            COEFF
          ≠BEGIN≠                                                        COEFF
            COMUDI2(B,N+1,EZ1P,N+1,T1,1).,                               COEFF
            COMUDI2(B,N+1,Z2P,-N-1,T2,1).,                               COEFF
            COMUDI2(B,N+1,Z1P,N+1,T3,1).,                                COEFF
            S1(/1/).=S1(/1/)+T1(/1/)., S1(/2/).=S1(/2/)+T1(/2/).,        COEFF
            S2(/1/).=S2(/1/)+T2(/1/)., S2(/2/).=S2(/2/)+T2(/2/).,        COEFF
            S3(/1/).=S3(/1/)+T3(/1/)., S3(/2/).=S3(/2/)+T3(/2/).,        COEFF
            EN(/1/).=EP(/1,N-K+1/)., EN(/2/).=EP(/2,N-K+1/).,            COEFF
            COMUDI1(S1,EN,LM2,-1).,                                      COEFF
            H.=K*(1/(N+.5)-1)+1., ≠COMMENT≠ RESP. 1 EN 1/(N+1/2).,       COEFF
            LM2(/1/).=LM2(/1/)+S3(/1/)., LM2(/2/).=LM2(/2/)+S3(/2/).,    COEFF
            COMUDI1(IZ12,LM2,LM2,1).,                                    COEFF
            LM3(/1/).=LM2(/1/)., LM3(/2/).=LM2(/2/)-FGAM(/2,I/).,        COEFF
            ≠FOR≠ J.=1,2 ≠DO≠                                            COEFF
            ≠BEGIN≠ LARRAY(/I,5,N,J/).=H*(1-2*K)*LM2(/J/).,              COEFF
                    LARRAY(/I,9,N,J/).=H*(1-2*K)*LM3(/J/).,              COEFF
                    LARRAY(/I,8,N,J/).=-S2(/J/).,                        COEFF
            ≠END≠.,                                                      COEFF
          ≠END≠.,                                                        COEFF
            R(/1/).=R(/2/).=0.,                                          COEFF
          ≠FOR≠ N.=0 ≠STEP≠ 1 ≠UNTIL≠ M ≠DO≠                             COEFF
          ≠BEGIN≠ COMUDI2(B,N,EZ2P,-N,T,1).,                             COEFF
            R(/1/).=R(/1/)+T(/1/)., R(/2/).=R(/2/)+T(/2/).,              COEFF
            LP3(/1/).=R(/1/)+S1(/1/)., LP3(/2/).=R(/2/)+S1(/2/).,        COEFF
            EN(/1/).=EP(/1,N+K/)., EN(/2/).=EP(/2,N+K/).,                COEFF
            COMUDI1(LP3,EN,LP3,1).,                                      COEFF
            ≠FOR≠ J.=1,2 ≠DO≠ LP3(/J/).=LP3(/J/)+LARRAY(/I,8,N,J/).,     COEFF
            COMUDI1(IZ12,LP3,LP3,1).,                                    COEFF
            H.=K*(1/(N+.5)-1)+1., ≠COMMENT≠ RESP. 1 EN 1/(N+1/2).,       COEFF
            ≠FOR≠ J.=1,2 ≠DO≠                                            COEFF
            ≠BEGIN≠ LP2(/J/).=LP3(/J/)+FGAM(/J,I/).,                     COEFF
              LARRAY(/I,4,N,J/).=H*LP2(/J/).,                            COEFF
              LARRAY(/I,8,N,J/).=H*LP3(/J/).,                            COEFF
            ≠END≠                                                        COEFF
          ≠END≠.,                                                        COEFF
      ≠END≠ I-CYCLE.,                                                    COEFF
      ≠COMMENT≠ COMPUTATION OF LAMBDA AND DLAMBDA/DN.,                   COEFF
      LN2.=LN(2.0).,LNZ1(/1/).=LN(ABSZ1).,LNZ1(/2/).=ARCTAN(Z1(/2/)/Z1(/1/)).,  COEFF
      ≠FOR≠ I.=1,2 ≠DO≠                                                  COEFF
      ≠BEGIN≠ SUM.=0., K.=1-I.,                                          COEFF
        ≠FOR≠ P.=0 ≠STEP≠ 1 ≠UNTIL≠ M+K ≠DO≠                             COEFF
        ≠BEGIN≠ ≠IF≠ P=0 ≠THEN≠                                          COEFF
          ≠BEGIN≠ FACT.=F(/1/).=1.0., F(/2/).=0.0 ≠END≠ ≠ELSE≠           COEFF
          ≠BEGIN≠ FACT.=(P+I-2.5)/P*FACT.,                               COEFF
            COHYPF(I-1.5,-P,-P-I+2.5).,                                  COEFF
            SUM.=SUM-.5/P/(2*P-1)                                        COEFF
          ≠END≠.,                                                        COEFF
          DCOHYPF(I-1.5,0,-P,-1,-P-I+2.5,-1).,                           COEFF
          D(/1/).=LNZ1(/1/)+2.0*(LN2+(K+1)/(2*P-1)+SUM).,                COEFF
```

```
      D(/2/).=LNZ1(/2/)., COMUDI1(F,D,D,1).,                          COEFF
      DB(/1,P/).=DF(/1/)-D(/1/)., DB(/2,P/).=DF(/2/)-D(/2/).,          COEFF
      B(/1,P/).=F(/1/)., B(/2,P/).=F(/2/)                               COEFF
    ≠END≠ P-CYCLE.,                                                     COEFF
      ≠FOR≠ J.=1,2 ≠DO≠                                                 COEFF
      ≠BEGIN≠ S1(/J/).=S2(/J/).=(K+1)*B(/J,0/).,                        COEFF
        DS1(/J/).=DS2(/J/).=(K+1)*DB(/J,0/).,                           COEFF
      ≠END≠.,                                                           COEFF
      ≠FOR≠ N.=1 ≠STEP≠ 1 ≠UNTIL≠ M ≠DO≠                                COEFF
      ≠BEGIN≠ COMUDI2(B,N+K,Z1P,N+K,T1,-1).,                            COEFF
        COMUDI2(B,N+K,EZ1P,N+K,T2,-1).,                                 COEFF
        COMUDI2(DB,N+K,Z1P,N+K,DT1,-1).,                                COEFF
        COMUDI2(DB,N+K,EZ1P,N+K,DT2,-1).,                               COEFF
        ≠FOR≠ J.=1,2 ≠DO≠                                               COEFF
        ≠BEGIN≠ S1(/J/).=S1(/J/)+T1(/J/)., DS1(/J/).=DS1(/J/)+DT1(/J/)., COEFF
          S2(/J/).=S2(/J/)+T2(/J/)., DS2(/J/).=DS2(/J/)+DT2(/J/).,      COEFF
          EN(/J/).=EP(/J,N/)                                            COEFF
        ≠END≠.,                                                         COEFF
        COMUDI1(EN,S2,L,1)., COMUDI1(EN,DS2,DNL,1).,                    COEFF
        H.=1.0+K*(1+1.0/N).,                                            COEFF
        ≠FOR≠ J.=1,2 ≠DO≠                                               COEFF
        ≠BEGIN≠ L(/J/).=S1(/J/)+L(/J/)., DNL(/J/).=DS1(/J/)+DNL(/J/).,  COEFF
          LARRAY(/I,1,N,J/).=H*L(/J/).,                                 COEFF
          LARRAY(/I,2,N,J/).=H*(K/N*L(/J/)+DNL(/J/))                    COEFF
        ≠END≠.,                                                         COEFF
      ≠END≠ N-CYCLE.,                                                   COEFF
  ≠END≠ I-CYCLE.,                                                       COEFF
      ≠FOR≠ I.=1,2 ≠DO≠ ≠FOR≠ K.=1,2,3,6,7,10,11 ≠DO≠ ≠FOR≠ J.=1,2 ≠DO≠  COEFF
      LARRAY(/I,K,0,J/).=0.0.,                                          COEFF
  SKIPF(43)., SKIPF(43)., BACKSPACE(43).,                               COEFF
    PUTARRAY(43,PARAM).,                                                COEFF
    PUTARRAY(43,LARRAY).,                                               COEFF
    ENDFILE(43).,  ENDFILE(43).,                                        COEFF
    ≠FOR≠ I.=1,2 ≠DO≠ ≠FOR≠ K.=1 ≠STEP≠ 1 ≠UNTIL≠ 11 ≠DO≠              COEFF
    ≠BEGIN≠ OUTPUT(41,≠(≠*,3S,8D,5B,3S,2ZD,/≠)≠,≠(≠I =≠)≠,I,≠(≠K =≠)≠,K)., COEFF
      ≠FOR≠ N.=0 ≠STEP≠ 1 ≠UNTIL≠ M ≠DO≠                                COEFF
      ≠BEGIN≠ OUTPUT(41,≠(≠/≠)≠).,                                      COEFF
        ≠FOR≠ P.=1,2 ≠DO≠ OUTPUT(41,≠(≠≠)≠,LARRAY(/I,K,N,P/))           COEFF
      ≠END≠                                                             COEFF
    ≠END≠                                                               COEFF
  ≠END≠                                                                 COEFF
  ≠END≠                                                                 COEFF
            ≠EOP≠                                                       COEFF
```

```
LINE    0      PROGRAM BEGINS        (MESSAGE)       1
LINE  331      PROGRAM ENDS          (MESSAGE)       1
LINE  331      SOURCE DECK ENDS      (MESSAGE)       1
LINE  131      NON-FORMAT STRING     (MESSAGE)       1
LINE  131      NON-FORMAT STRING     (MESSAGE)       1
LINE  131      NON-FORMAT STRING     (MESSAGE)       1
LINE  132      NON-FORMAT STRING     (MESSAGE)       1
LINE  132      NON-FORMAT STRING     (MESSAGE)       1
LINE  132      NON-FORMAT STRING     (MESSAGE)       1
LINE  132      NON-FORMAT STRING     (MESSAGE)       1
LINE  133      NON-FORMAT STRING     (MESSAGE)       1
LINE  133      NON-FORMAT STRING     (MESSAGE)       1
LINE  133      NON-FORMAT STRING     (MESSAGE)       1
LINE  133      NON-FORMAT STRING     (MESSAGE)       1
LINE  134      NON-FORMAT STRING     (MESSAGE)       1
LINE  134      NON-FORMAT STRING     (MESSAGE)       1
LINE  134      NON-FORMAT STRING     (MESSAGE)       1
LINE  324      NON-FORMAT STRING     (MESSAGE)       1
LINE  324      NON-FORMAT STRING     (MESSAGE)       1

         THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE      I,L,O,Q,X


CORE MAP  14.37.30.  NORMAL          CONTROL                                       000100    033462    031262    002200
          ---TIME---LOAD MODE --L1--L2-----TYPE---------------USER---+----CALL-------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
          FWA LOADER  050741  FWA TABLES  046346
          -PROGRAM----ADDRESS-                      --LABELED---COMMON--
          XXALGOL     000340                                  DATA      000100
          ALGORUN     013340                                  DATA      000100

          ALGLB00     015712                                  DATA      000100
          ALGLB01     021127                                  DATA      000100
          ALGLB02     021656                                  DATA      000100
          ALGLB03     027525                                  DATA      000100
          ALGLB06     030360                                  DATA      000100
          --ENTRY-----ADDRESS-                                        REFERENCES
          XXALGOL     012375
          ALGORUN     013340          XXALGOL

          ALGLB00     015712          XXALGOL

          ALGLB01     021127          XXALGOL

          ALGLB02     021656          XXALGOL

          ALGLB03     027530          XXALGOL

          ALGLB06     030360          XXALGOL

          ----UNSATISFIED EXTERNALS-----                              REFERENCES


CHANNEL,60=INPUT,P80,R
CHANNEL,61=OUTPUT,P136,PP60,R
CHANNEL,40=60
CHANNEL,41=61
CHANNEL,43=LU43,A,B
CHANNEL,END
```

```
≠BEGIN≠ ≠COMMENT≠ INTEGRATIECONSTANTEN T. B. V. PROGR. GAMMA-FLOW 4.,
        ≠REAL≠ PI,LN4,EPSZERO,ALFA,GAMMA,Z1ABS,Z2ABS,TI,VTI,FMINTI,
        FTI,CXTI,CXPI,CYTI,CYPI,FACTOR,C3Y,XTI,XPI,YTI,YPI,LNEPS.,
        ≠INTEGER≠ N,K,D,I,P,J,SIG,SIDE,RECO,A,AANTAL,CASE.,

        ≠ARRAY≠ Z1,Z2,Z1DZ2,EPS,LNZ1,FGAM,B,DB,D2B,L,DL,D2L,
        KL1,KL2,KL3,Z1P(/1..2/),
        PSITI(/0..140,1..5,0..1/),
        DX,DY(/1..3,1..2,0..1/),SIG1,SIG2,SIG3(/1..2,1..2/),
        LARRAY(/1..2,1..11,0..100,1..2/).,

        ≠PROCEDURE≠ COMUDI(A,B,C,T).,≠INTEGER≠ T.,≠ARRAY≠ A,B,C.,
        ≠BEGIN≠ ≠REAL≠ MOD.,≠ARRAY≠ P,Q(/1..2/).,
        P(/1/).=A(/1/).,P(/2/).=A(/2/).,Q(/1/).=B(/1/).,Q(/2/).=B(/2/).,
        ≠IF≠ T ≠GREATER≠ 0 ≠THEN≠ MOD.=1 ≠ELSE≠ MOD.=
        1/(Q(/1/)*Q(/1/)+Q(/2/)*Q(/2/)).,
        C(/1/).=(P(/1/)*Q(/1/)-T*P(/2/)*Q(/2/))*MOD.,

        C(/2/).=(T*P(/1/)*Q(/2/)+P(/2/)*Q(/1/))*MOD.,
        ≠END≠ COMUDI.,

        ≠PROCEDURE≠ COHYPF(A,B,C,Z,F).,≠VALUE≠A,B,C.,
        ≠REAL≠ A,B,C., ≠ARRAY≠ Z,F.,
        ≠BEGIN≠ ≠REAL≠PART.,≠INTEGER≠K,KM.,≠ARRAY≠TERM,TERMH(/1..2/).,
                F(/1/).=TERM(/1/).=1.,TERM(/2/).=F(/2/).=0., KM.=0.,
        AA..    K.=KM+1.,PART.=(A+KM)*(B+KM)/((C+KM)*K).,
                COMUDI(TERM,Z,TERMH,1).,TERM(/1/).=PART*TERMH(/1/).,
                TERM(/2/).=PART*TERMH(/2/).,
                F(/1/).=F(/1/)+TERM(/1/).,F(/2/).=F(/2/)+TERM(/2/).,
                KM.=K.,   ≠IF≠ ABS(TERM(/1/)/F(/1/)) ≠LESS≠ ≠-8  ≠AND≠

                ABS(TERM(/2/)/F(/2/)) ≠LESS≠≠-8  ≠THEN≠ ≠GOTO≠ BB
                ≠ELSE≠ ≠GOTO≠ AA.,
   BB..  ≠END≠ COHYPF.,

        ≠PROCEDURE≠ DCOHYPF(A,DA,B,DB,C,DC,Z,DF).,≠VALUE≠A,DA,B,DB,
        C,DC.,≠REAL≠A,DA,B,DB,C,DC.,≠ARRAY≠ Z,DF.,
        ≠BEGIN≠≠REAL≠DTERM.,≠INTEGER≠K.,≠ARRAY≠KLAD,TERM,T,DT(/1..2/).,
                DF(/1/).=DF(/2/).=DT(/1/).=DT(/2/).=TERM(/2/).=0.,
                T(/2/).=0., T(/1/).=1., K.=-1.,
        AA..    K.=K+1.,TERM(/1/).=(A+K)*(B+K)/(C+K)/(K+1).,
                DTERM.=(((B+K)*DA+(A+K)*DB)/(K+1)-TERM(/1/)*DC)/(C+K).,
                KLAD(/1/).=DTERM*T(/1/)+TERM(/1/)*DT(/1/).,
                KLAD(/2/).=DTERM*T(/2/)+TERM(/1/)*DT(/2/).,
                COMUDI(KLAD,Z,DT,1).,DF(/1/).=DF(/1/)+DT(/1/).,
                DF(/2/).=DF(/2/)+DT(/2/).,  ≠IF≠ ABS(DT(/1/)/DF(/1/))
                ≠GREATER≠≠-8 ≠OR≠ABS(DT(/2/)/DF(/2/))≠GREATER≠≠-8

                ≠THEN≠≠BEGIN≠ COMUDI(TERM,Z,KLAD,1).,
                COMUDI(KLAD,T,T,1).,≠GOTO≠ AA ≠END≠
        ≠END≠ DCOHYPF.,

        ≠PROCEDURE≠ D2COHYPF(A,DA,D2A,B,DB,D2B,C,DC,D2C,Z,D2F).,
        ≠VALUE≠ A,DA,D2A,B,DB,D2B,C,DC,D2C.,
        ≠REAL≠ A,DA,D2A,B,DB,D2B,C,DC,D2C., ≠ARRAY≠ Z,D2F.,
        ≠BEGIN≠ ≠REAL≠ DAB,DAC,DBC,DCC,AK,BK,CK,AB,AC,BC,CC,FK,F1K,F2K.,
                ≠INTEGER≠ J,K,K1.,  ≠ARRAY≠ T,DT,D2T,KLAD(/1..2/).,
                T(/1/).=1., K.=-1., D2F(/1/).=D2F(/2/).=T(/2/).=
                DT(/1/).=DT(/2/).=D2T(/1/).=D2T(/2/).=0.,
                DAB.=DA*DB., DAC.=DA*DC., DBC.=DB*DC., DCC.=DC*DC.,
        AA..    K.=K+1., K1.=K+1., AK.=A+K., BK.=B+K., CK.=C+K.,

                AB.=AK*BK., AC.=AK*CK., BC.=BK*CK., CC.=CK*CK.,
```

APPENDIX B

LISTING OF INTCONS

```
                      FK.=AB/(K1*CK)., F1K.=(BC*DA+AC*DB-AB*DC)/(K1*CC).,
         F2K.=(BC*D2A+AC*D2B-AB*D2C)/(K1*CC)+2*(CC*DAB-BC*DAC-AC*DBC+AB*DCC)/
                      (K1*CK*CC).,  #FOR# J.=1,2 #DO# KLAD(/J/).=F2K*T(/J/)+
                      2*F1K*DT(/J/)+FK*D2T(/J/).,   COMUDI(KLAD,Z,D2T,1).,
                      #FOR# J.=1,2 #DO# D2F(/J/).=D2F(/J/)+D2T(/J/).,
                      #IF# ABS(D2T(/1/)/D2F(/1/)) #GREATER# #-8  #OR#
                      ABS(D2T(/2/)/D2F(/2/)) #GREATER# #-8  #THEN#
                      #BEGIN# #FOR# J.=1,2 #DO# KLAD(/J/).=F1K*T(/J/)+FK
                      *DT(/J/).,  COMUDI(Z,KLAD,DT,1).,  #FOR# J.=1,2 #DO#

                      KLAD(/J/).=FK*T(/J/)., COMUDI(Z,KLAD,T,1).,  #GOTO# AA
                      #END#
               #END# D2COHYPF.,

         #PROCEDURE# TAPE(T,AR)., #VALUE# T., #REAL# T., #ARRAY# AR.,
         #BEGIN# #ARRAY# H(/1..4/)., #INTEGER# KT,KH.,
           KT.=1200*T.,
         READ..  GETARRAY(43,AR).,  KH.=600/AR(/0,2,1/).,
           #IF# KH #NOTEQUAL# KT #THEN# #GOTO# READ..
         #END#.,

         #PROCEDURE# CONSTANTS.,

           #BEGIN#
           #REAL# H,CZ1,CZ2,MOD,ARC.,
           #ARRAY# EPS1,H1,H2(/1..2/).,

           #PROCEDURE# CMD(A,B,Z,T)., #VALUE# T., #REAL# T., #ARRAY# A,B,Z.,
           #BEGIN##REAL# A1,A2,B1,B2,MOD.,
             A1.=A(/1/)., A2.=A(/2/)., B1.=B(/1/)., B2.=B(/2/).,
             MOD.=#IF# 0 #LESS# T #THEN# 1.0 #ELSE# 1.0/(B1*B1+B2*B2).,
             Z(/1/).=MOD*(A1*B1-T*A2*B2).,
             Z(/2/).=MOD*(A2*B1+T*A1*B2).,
           #END# CMD.,

           PI.=4.0*ARCTAN(1.0).,
           EPS(/1/).=EPSZERO*COS(2.0*ALFA).,
           EPS1(/1/).=1.0+EPS(/1/).,
           EPS(/2/).=EPS1(/2/).=EPSZERO*SIN(2.0*ALFA).,
           CMD(EPS,EPS1,H1,-1).,
           CMD(H1,EPS1,H2,-1).,
           H.=4.0-(0.5*GAMMA/PI) #POWER# 2.,
           CZ1.=1.0-H*H2(/1/)., CZ2.=-H*H2(/2/).,
           MOD.=(CZ1*CZ1+CZ2*CZ2) #POWER# 0.25.,
           ARC.=0.5*(ARCTAN(CZ2/CZ1)+(#IF# CZ1 #LESS# 0 #THEN# SIGN(CZ2)*PI
             #ELSE# 0.0)).,

           H2(/1/).=0.5*(1.0-MOD*COS(ARC)).,
           H2(/2/).=-0.5*MOD*SIN(ARC).,
           CMD(H2,H1,Z1,-1).,
           H2(/1/).=1.0-H2(/1/).,
           H2(/2/).=-H2(/2/).,
           CMD(H2,H1,Z2,-1).,
           Z1ABS.=SQRT(Z1(/1/) #POWER# 2+Z1(/2/) #POWER# 2).,
           Z2ABS.=SQRT(Z2(/1/) #POWER# 2+Z2(/2/) #POWER# 2).,
           CMD(Z1,Z2,Z1DZ2,-1).,
           #END# PROCEDURE CONSTANTS.,

           INPUT(40,#(##)#,CASE,EPSZERO,ALFA,GAMMA).,

           CONSTANTS.,
           OUTPUT(61,#(#2(/,7(19S))#)#,#(#CASE#)#,#(#EPSZERO#)#,#(#ALFA#)#,
```

```
     ≠(≠GAMMA≠)≠,≠(≠RE ZETA1≠)≠,≠(≠IM ZETA1≠)≠,≠(≠RE ZETA2≠)≠,          INTCONS
     ≠(≠IM ZETA2≠)≠,≠(≠ABSZETA1≠)≠,≠(≠ABSZETA2≠)≠,≠(≠RE Z1 DIV Z2≠)≠,   INTCONS
     ≠(≠IM Z1 DIV Z2≠)≠,≠(≠RE EPS≠)≠,≠(≠IM EPS≠)≠).,                     INTCONS
   OUTPUT(61,≠(≠2(/,7(+D.10D≠+ZD2B))≠)≠,                                INTCONS
     CASE,EPSZERO,ALFA,GAMMA,Z1(/1/),Z1(/2/),Z2(/1/),                   INTCONS
     Z2(/2/),Z1ABS,Z2ABS,Z1DZ2(/1/),Z1DZ2(/2/),EPS(/1/),                INTCONS
     EPS(/2/)).,                                                        INTCONS
 LNZ1(/1/).=LN(Z1ABS)., LNZ1(/2/).= ARCTAN(Z1(/2/)/Z1(/1/)).,           INTCONS

                                                                        INTCONS
   ≠BEGIN≠ ≠ARRAY≠ PARAM(/1..6/).,                                      INTCONS
   SKIPF(43).,                                                          INTCONS
   EOF(43,ALARM)., ≠GOTO≠ SEARCH.,                                      INTCONS
 ALARM..                                                                INTCONS
   OUTPUT(41,≠(≠//,≠(≠CASE UNKNOWN ON TAPE≠)≠≠)≠)., ≠GOTO≠ EOP.,        INTCONS
 SEARCH..                                                               INTCONS
   GETARRAY(43,PARAM)., GETARRAY(43,LARRAY).,                           INTCONS
   ≠IF≠ ABS(CASE-PARAM(/3/))≠GREATER≠0.5 ≠OR≠ ABS(EPSZERO-PARAM(/4/))   INTCONS
   ≠GREATER≠ ≠-8 ≠OR≠ ABS(ALFA-PARAM(/5/)) ≠GREATER≠ ≠-8 ≠OR≠ ABS(GAMMA- INTCONS
   PARAM(/6/)) ≠GREATER≠ ≠-8 ≠THEN≠ ≠GOTO≠ SEARCH.,                     INTCONS
   ≠END≠.,                                                              INTCONS

   REWIND(43).,                                                         INTCONS
 OUTPUT(41,≠(≠//,≠(≠NUMBERS OF TERMS IN POWER SERIES≠)≠≠)≠).,           INTCONS
 ≠FOR≠ I.=1,2 ≠DO≠ ≠FOR≠ P.=1,2 ≠DO≠ ≠BEGIN≠                            INTCONS
 SIG1(/I,P/).=LARRAY(/I,3,1,P/).. SIG3(/I,P/).=LARRAY(/I,7,1,P/) ≠END≠.. INTCONS
 RECO.=0., LN4.=LN(4).,  LNEPS.=-LN(EPSZERO).,                          INTCONS
 GAMMA.= .5*GAMMA/PI.,                                                  INTCONS
 FGAM(/1/).=SQRT(1-.25*GAMMA*GAMMA)., FGAM(/2/).=-.25*GAMMA*GAMMA/      INTCONS
 FGAM(/1/).,                                                            INTCONS
 INPUT(40,≠(≠≠)≠,AANTAL).,                                              INTCONS
 ≠FOR≠ A.=1 ≠STEP≠ 1 ≠UNTIL≠ AANTAL ≠DO≠                                INTCONS
 ≠BEGIN≠ INPUT(40,≠(≠≠)≠,TI)., TAPE(TI,PSITI).,  VTI.=SQRT(TI).,        INTCONS
 FMINTI.=-1/VTI+1.25*(3.8*(PSITI(/1,2,0/)+2*TI*PSITI(/1,2,1/))+         INTCONS

 2.8*PSITI(/1,1,0/)-2*VTI*(1-TI)≠POWER≠2.5*(2.4*LN(TI)+2.8)).,          INTCONS
 FTI.=2.5*TI*(1-TI)≠POWER≠2.5.,                                         INTCONS
             CXTI.=CXPI.=CYTI.=CYPI.=0.,                                INTCONS
 ≠FOR≠ I.=1,2 ≠DO≠ ≠BEGIN≠ ≠IF≠ I ≠EQUAL≠ 1 ≠THEN≠                      INTCONS
 ≠BEGIN≠ DCOHYPF(-.5,0,0,-1,1.5,-1,Z1DZ2,DB).,                          INTCONS
 D2COHYPF(-.5,0,0,0,-1,0,1.5,-1,0,Z1DZ2,D2B)., L(/1/).=1+EPS(/1/).,     INTCONS
 L(/2/).=EPS(/2/).,  KL1(/1/).=-LNZ1(/1/)-LN4+2+DB(/1/).,               INTCONS
 KL1(/2/).=-LNZ1(/2/)+DB(/2/).,  COMUDI(L,KL1,DL,1).,                   INTCONS
 KL1(/1/).=-LNZ1(/1/)-LN4+2., KL1(/2/).=-LNZ1(/2/).,                    INTCONS
 COMUDI(KL1,DB,KL2,1)., COMUDI(KL1,KL1,KL1,1).,                         INTCONS
 KL1(/1/).=KL1(/1/)+2*KL2(/1/)+PI*PI/3+4+D2B(/1/).,                     INTCONS
 KL1(/2/).=KL1(/2/)+2*KL2(/2/)+D2B(/2/)., COMUDI(L,KL1,D2L,1).,         INTCONS
 COHYPF(-.5,-1,.5,Z1DZ2,B)., DCOHYPF(-.5,0,-1,-1,.5,-1,Z1DZ2,DB).,      INTCONS

 D2COHYPF(-.5,0,0,-1,-1,0,.5,-1,0,Z1DZ2,D2B)., ≠FOR≠ K.=1,2 ≠DO≠        INTCONS
 ≠BEGIN≠ B(/K/).=-.5*B(/K/).,DB(/K/).=-.5*DB(/K/).,                     INTCONS
 D2B(/K/).=-.5*D2B(/K/) ≠END≠., COMUDI(B,Z1,KL1,-1).,                   INTCONS
 L(/1/).=L(/1/)+2*KL1(/1/).,  L(/2/).=L(/2/)+2*KL1(/2/).,               INTCONS
 KL1(/1/).=-LNZ1(/1/)-LN4-1., KL1(/2/).=-LNZ1(/2/).,COMUDI(KL1,B,KL2,1)., INTCONS
 KL2(/1/).=KL2(/1/)+DB(/1/)., KL2(/2/).=KL2(/2/)+DB(/2/).,              INTCONS
 COMUDI(KL2,Z1,KL2,-1)., DL(/1/).=DL(/1/)+2*KL2(/1/).,DL(/2/).=DL(/2/)+ INTCONS
 2*KL2(/2/)., COMUDI(KL1,KL1,KL2,1)., COMUDI(KL2,B,KL2,1).,             INTCONS
 COMUDI(KL1,DB,KL1,1).,                                                 INTCONS
 KL1(/1/).=KL2(/1/)+2*KL1(/1/)+(PI*PI/3+1)*B(/1/)+D2B(/1/).,            INTCONS
 KL1(/2/).=KL2(/2/)+2*KL1(/2/)+(PI*PI/3+1)*B(/2/)+D2B(/2/).,            INTCONS
 COMUDI(KL1,Z1,KL1,-1)., D2L(/1/).=D2L(/1/)+2*KL1(/1/).,                INTCONS
 D2L(/2/).=D2L(/2/)+2*KL1(/2/).,   ≠END≠ ≠ELSE≠                         INTCONS
```

```
#BEGIN# DCOHYPF(.5,0,0,-1,.5,-1,Z1DZ2,DB).,                                 INTCONS
D2COHYPF(.5,0,0,0,-1,0,.5,-1,0,Z1DZ2,D2B).,  L(/1/).=-1-EPS(/1/).,          INTCONS
L(/2/).= -EPS(/2/)., KL1(/1/).=-LNZ1(/1/)-LN4+DB(/1/).,                     INTCONS
KL1(/2/).=-LNZ1(/2/)+DB(/2/)., COMUDI(L,KL1,KL1,1).,                        INTCONS
DL(/1/).=-L(/1/)+KL1(/1/)., DL(/2/).= -L(/2/)+KL1(/2/).,                    INTCONS
KL1(/1/).=-LNZ1(/1/)-LN4., KL1(/2/).=-LNZ1(/2/).,                           INTCONS
COMUDI(KL1,DB,KL2,1)., COMUDI(KL1,KL1,KL1,1).,                              INTCONS
KL1(/1/).=KL1(/1/)+2*KL2(/1/)+PI*PI/3+D2B(/1/).,                            INTCONS
KL1(/2/).=KL1(/2/)+2*KL2(/2/)+D2B(/2/)., COMUDI(L,KL1,KL1,1).,              INTCONS
D2L(/1/).= -2*DL(/1/)+KL1(/1/)., D2L(/2/).= -2*DL(/2/)+KL1(/2/) #END#.,     INTCONS
XPI.=FGAM(/I/)*PI*FTI*DL(/2/).,                                             INTCONS
XTI.=FGAM(/I/)*(VTI*FMINTI*DL(/1/)+.5*FTI*D2L(/1/)).,                       INTCONS
YPI.=-FGAM(/I/)*PI*FTI*DL(/1/).,                                            INTCONS

YTI.=FGAM(/I/)*(VTI*FMINTI*DL(/2/)+.5*FTI*D2L(/2/)).,                       INTCONS
CXTI.=CXTI+XTI.,CXPI.=CXPI+XPI.,CYTI.=CYTI+YTI.,CYPI.=CYPI+YPI #END#.,      INTCONS
#FOR# I.=1,2 #DO#                                                           INTCONS
#BEGIN# #IF# A=1 #THEN#                                                     INTCONS
#BEGIN# #REAL# ARG,EMR,EMI,EM,M,TOL,SOM,T1,T2.,                             INTCONS
TOL.=(1.0/Z1ABS-1.0)*#-7.,                                                  INTCONS
Z1P(/1/).=1., Z1P(/2/).=DL(/1/).=DL(/2/).=SOM.=0.0.,                        INTCONS
P.=1-I.,                                                                    INTCONS
NEXTP..  P.=P+1., COMUDI(Z1P,Z1,Z1P,1).,                                    INTCONS
COHYPF(I-1.5,P+I-1,P+1.5,Z1DZ2,B).,                                         INTCONS
DCOHYPF(I-1.5,0,P+I-1,-1,P+1.5,-1,Z1DZ2,DB).,                               INTCONS
FACTOR.=#IF#P #EQUAL# 2-I #THEN# (8*I-10)/3 #ELSE# (P+I-2)/(P+.5)          INTCONS

*FACTOR.,  #FOR# K.=1,2 #DO# #BEGIN# B(/K/).=FACTOR*B(/K/).,                INTCONS
DB(/K/).=FACTOR*DB(/K/) #END#.,                                             INTCONS
SOM.=SOM+1/(P+I-1)/(2*(P+I-1)-1).,                                          INTCONS
KL1(/1/).=-LNZ1(/1/)-LN4+(#IF# I=1 #THEN# 1/P+2/(2*P+1) #ELSE#              INTCONS
1/(P+1))+SOM., KL1(/2/).=-LNZ1(/2/).,                                       INTCONS
COMUDI(KL1,B,KL1,1)., KL1(/1/).=KL1(/1/)+DB(/1/)., KL1(/2/).=KL1(/2/)+      INTCONS
DB(/2/)., COMUDI(KL1,Z1P,KL1,1).,                                           INTCONS
M.=P+I., #IF# M #GREATER# 690/LNEPS #THEN#                                  INTCONS
#BEGIN# EMR.=EMI.=0 #END# #ELSE#                                            INTCONS
#BEGIN# EM.=EPSZERO#POWER#M., ARG.=2*M*ALFA.,                               INTCONS
EMR.=EM*COS(ARG)., EMI.=EM*SIN(ARG)                                         INTCONS

#END#.,                                                                     INTCONS
T1.=(1+EMR)*KL1(/1/)-EMI*KL1(/2/)., DL(/1/).=DL(/1/)+T1.,                   INTCONS
T2.=(1+EMR)*KL1(/2/)+EMI*KL1(/1/)., DL(/2/).=DL(/2/)+T2.,                   INTCONS
#IF# (T1*T1+T2*T2)/(DL(/1/)*DL(/1/)+DL(/2/)*DL(/2/))#GREATER#               INTCONS
TOL*TOL #THEN##GOTO# NEXTP.,                                                INTCONS
OUTPUT(41,#(#//,3ZD2B#)#,P).,                                               INTCONS
#FOR# K.=1,2 #DO# SIG2(/I,K/).=DL(/K/)+(1-I)*SIG1(/I,K/)                    INTCONS
#END# A=1.,                                                                 INTCONS
L(/1/).= SIG1(/I,1/)., L(/2/).=SIG1(/I,2/).,                                INTCONS
DL(/1/).=SIG2(/I,1/).,  DL(/2/).=SIG2(/I,2/).,                              INTCONS
XPI.=FGAM(/I/)*PI*FTI*L(/2/)., YPI.=-FGAM(/I/)*PI*FTI*L(/1/).,              INTCONS
XTI.=FGAM(/I/)*(VTI*FMINTI*L(/1/)+FTI*DL(/1/)).,                            INTCONS

YTI.=FGAM(/I/)*(VTI*FMINTI*L(/2/)+FTI*DL(/2/)).,                            INTCONS
CXTI.=CXTI+XTI., CXPI.=CXPI+XPI., CYTI.=CYTI+YTI., CYPI.=CYPI+YPI.,         INTCONS
L(/1/).= SIG3(/I,1/)., L(/2/).= SIG3(/I,2/).,                               INTCONS
XTI.=FGAM(/I/)*L(/1/)*.5*FTI., YTI.=-FGAM(/I/)*L(/2/)*.5*FTI.,              INTCONS
CXTI.=CXTI+XTI., CYTI.=CYTI+YTI #END#.,                                     INTCONS
C3Y.=.5*GAMMA*(VTI*FMIN TI-1.25*VTI*(PSITI(/1,2,0/)+2*TI*PSITI(/1,2,1/))    INTCONS
+2.8*FTI).,                                                                 INTCONS
DX(/1,1,0/).=-CXTI., DX(/1,1,1/).=CXTI., DY(/1,1,0/).= -CYTI.,              INTCONS
DY(/1,1,1/).=CYTI.,  #FOR# J.=2,3 #DO# #FOR# SIG.=0,1 #DO#                  INTCONS
#BEGIN# DX(/J,1,SIG/).= -CXPI., DY(/J,1,SIG/).=-CYPI #END#.,                INTCONS
```

```
       ≠FOR≠ J.=1,2,3 ≠DO≠ ≠FOR≠ SIG.=0,1 ≠DO≠ ≠BEGIN≠ DX(/J,2,SIG/).=

       -DX(/J,1,SIG/)., DY(/J,2,SIG/).=-DY(/J,1,SIG/) ≠END≠.,
       ≠FOR≠ SIDE.=1,2 ≠DO≠ ≠FOR≠ SIG.=0,1 ≠DO≠ DY(/3,SIDE,SIG/).=
       DY(/3,SIDE,SIG/)-C3Y.,
       OUTPUT(41,≠(≠///,5S,D.4D,/≠)≠,≠(≠TAU1=≠)≠,TI).,
       ≠FOR≠ SIG.=0,1 ≠DO≠ ≠BEGIN≠ OUTPUT(41,≠(≠/≠)≠).,
       ≠FOR≠ J.=1,2,3 ≠DO≠ ≠BEGIN≠ OUTPUT(41,≠(≠/≠)≠).,
       ≠FOR≠ SIDE.=1,2 ≠DO≠ OUTPUT(41,≠(≠2(+ZD.6D2B)≠)≠,DX(/J,SIDE,SIG/),
240**  DY(/J,SIDE,SIG/)).,
       ≠END≠ ≠END≠ ≠END≠., EOP.. ≠END≠
             ≠EOP≠
```

```
LINE    0          PROGRAM BEGINS        (MESSAGE)           1
LINE  241          PROGRAM ENDS          (MESSAGE)           1
LINE  241          SOURCE DECK ENDS      (MESSAGE)           1
LINE  115          NON-FORMAT STRING     (MESSAGE)           1
LINE  115          NON-FORMAT STRING     (MESSAGE)           1
LINE  115          NON-FORMAT STRING     (MESSAGE)           1
LINE  116          NON-FORMAT STRING     (MESSAGE)           1
LINE  116          NON-FORMAT STRING     (MESSAGE)           1
LINE  116          NON-FORMAT STRING     (MESSAGE)           1
LINE  116          NON-FORMAT STRING     (MESSAGE)           1
LINE  117          NON-FORMAT STRING     (MESSAGE)           1
LINE  117          NON-FORMAT STRING     (MESSAGE)           1
LINE  117          NON-FORMAT STRING     (MESSAGE)           1
LINE  117          NON-FORMAT STRING     (MESSAGE)           1
LINE  118          NON-FORMAT STRING     (MESSAGE)           1
LINE  118          NON-FORMAT STRING     (MESSAGE)           1
LINE  118          NON-FORMAT STRING     (MESSAGE)           1
LINE  236          NON-FORMAT STRING     (MESSAGE)           1

         THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE      I,L,O,Q,X


CORE MAP  17.56.06.  NORMAL            CONTROL                                                000100    031421    027221    002200
    ---TIME---LOAD MODE --L1--L2-----TYPE---------------USER---+----CALL-------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
    FWA LOADER  053741  FWA TABLES  051363
    -PROGRAM----ADDRESS-                          --LABELED---COMMON--
    XXALGOL     000340                                    DATA        000100
    ALGORUN     011340                                    DATA        000100

    ALGLB00     013712                                    DATA        000100
    ALGLB01     017127                                    DATA        000100
    ALGLB02     017656                                    DATA        000100
    ALGLB05     025525                                    DATA        000100
    ALGLB06     026317                                    DATA        000100
    --ENTRY-----ADDRESS-                                          REFERENCES
    XXALGOL     010371
    ALGORUN     011340          XXALGOL

    ALGLB00     013712          XXALGOL

    ALGLB01     017127          XXALGOL

    ALGLB02     017656          XXALGOL

    ALGLB05     025525          XXALGOL

    ALGLB06     026317          XXALGOL

    ----UNSATISFIED EXTERNALS-----                                REFERENCES


CHANNEL,60=INPUT,P80,R
CHANNEL,61=OUTPUT,P136,PP60,R
CHANNEL,40=60
CHANNEL,41=61
CHANNEL,43=LU43,A,8,05
CHANNEL,END
```

```
'BEGIN' 'COMMENT' THE COMPUTATION OF TAU(C) AND THETA(C)..
'INTEGER' MAX,CASE..
MAX.=100..
'BEGIN' 'REAL' EPSZERO,ALFA,GAMMA,TI,T,TH,LNEPSO,TII,TI,SQTI,SQTII,TV,
THC,
INTEG,QTI,QT,FT,PHI,TOL,ZERO,LC,PHI2,PHIT,PHITH,PI,RAD,DTAU,DTHETA..
'INTEGER' K,II,N,J,D,TYPE,NI,INT,FNR,TINR,SLRI,LNR,R,I,RI,S,TNR..
'ARRAY' FACT,FGAM(/1..3/),Q(/1..48/),LARRAY(/1..2,1..11,0..100,1..2/),
PSIT,PSITI(/0..140,1..5,0..1/),C(/1..MAX/),DPDT,DPDTH,LOC(/-14..48/),
B(/1..2,0..1/),DUM,LA(/1..2/)..
'INTEGER' 'ARRAY' CO,SIGNL(/1..2/),ON1(/-14..-1/),LMAX(/1..5,1..48/)..

'PROCEDURE' CHAPLYGIN(TAU,PSIT).. 'REAL' TAU.. 'ARRAY' PSIT..
'BEGIN' 'COMMENT' CHAPLYGINFUNCTIONS FOR TAU LESS .05 ..
'REAL' TOL,    H,T,DTT,DNT,DTDNT,SOM1,SOM2,SOM3,SOM4,F1K,F2K,
PSI,DTPSI,DNPSI,DTDNPSI,ST,SDTT..
'INTEGER' M,K,A,B,I,U..
'REAL''PROCEDURE'N..N.=A*M+.5*B..
    TOL.='-11..
    PSIT(/0,3,0/).=PSIT(/0,3,1/).=0..
PSIT(/1,3,0/).=TAU'POWER'(-.5).. PSIT(/1,3,1/).=-.5*TAU'POWER'(-1.5)..
'FOR' M.=0 'STEP' 1 'UNTIL' 100 'DO'
'BEGIN' A.=1.. B.=0..
AA.. T.=SOM1.=1.. I.=(-7*A+6*B+9)/2..
DTT.=SOM2.=DNT.=SOM3.=DTDNT.=SOM4.=0..
'IF'M'EQUAL'0'AND'B'EQUAL'0'THEN''GOTO'CC.. K.=0..
BB.. K.=K+1.. F1K.=(-1.25*N*(N+1)+(K-1)*(N+K-3.5))/(N+K)/K*TAU..
'IF' B'EQUAL' 0 'THEN'
'BEGIN'F2K.=1.25/K*(-1+(K-1)/(N+K)'POWER'2*(K+2.8))*TAU..
DTDNT.=F1K*(DTDNT+DNT/TAU)+F2K*(DTT+T/TAU)..
DNT.=F1K*DNT+F2K*T.. SOM3.=SOM3+DNT.. SOM4.=SOM4+DTDNT 'END'..
DTT.=F1K*(DTT+T/TAU).. T.=F1K*T.. SOM1.=SOM1+T.. SOM2.=SOM2+DTT..
'IF' ABS(SOM1*SOM2) 'LESS' '-300 'THEN' 'GOTO' BB..
'IF' ABS(T/SOM1) 'GREATER' TOL 'OR' ABS(DTT/SOM2) 'GREATER' TOL
'THEN' 'GOTO' BB..  'IF' B 'NOTEQUAL' 0 'THEN' 'GOTO' CC..
'IF' ABS(SOM3*SOM4) 'LESS' '-300 'THEN' 'GOTO' BB..
'IF' ABS(DNT/SOM3) 'GREATER' TOL 'OR' ABS(DTDNT/SOM4) 'GREATER' TOL
'THEN' 'GOTO' BB..
CC..H.=TAU'POWER'(.5*N).. PSIT(/M,I,0/).=PSI.=H*SOM1..
PSIT(/M,I,1/).=DTPSI.=.5*N/TAU*PSI+H*SOM2.. 'IF' B'EQUAL' 0 'THEN'
'BEGIN' PSIT(/M,2,0/).=DNPSI.=.5*LN(TAU)*PSI+H*SOM3..
PSIT(/M,2,1/).=DTDNPSI.=.5*LN(TAU)*DTPSI+.5/TAU*PSI+H*(.5*N/TAU*SOM3+
  SOM4)..
B.=1.. 'GOTO' AA 'END'..
'IF' B'EQUAL'1 'THEN' 'BEGIN' A.=-1.. B.=-1.. 'GOTO' AA 'END'..
'IF' M 'GREATER' 1 'THEN'
'BEGIN' K.=0.. T.=ST.=1.. DTT.=SDTT.=DNT.=DTDNT.=0..
SOM1.=SOM2.=SOM3.=SOM4.=0..
DD.. K.=K+1.. 'IF' K 'NOTEQUAL' M 'THEN'
'BEGIN' F1K.=(-1.25*M*(M-1)+(K-1)*(K-M-3.5))/(K-M)/K*TAU..
F2K.=1.25/K*(-1+(K-1)/(K-M)/(K-M)*(K+2.8))*TAU 'END' 'ELSE'
'BEGIN'F1K.=1.25*(1-M)/M*(M+2.8)*TAU..
  F2K.=(3.5-2.25/M)*TAU 'END'..
DTDNT.=F1K*(DTDNT+DNT/TAU)+F2K*(DTT+T/TAU).. DNT.=F1K*DNT+F2K*T..
DTT.=F1K*(DTT+T/TAU).. T.=F1K*T..
'IF' K'LESS'M'THEN''BEGIN'ST.=ST+T.. SDTT.=SDTT+DTT..
'IF' ABS(ST*SDTT) 'GREATER' '-300 'THEN' 'BEGIN'
'IF' ABS(T/ST) 'LESS' TOL  'AND' ABS(DTT/SDTT) 'LESS' TOL
```

APPENDIX C
LISTING OF SADDPNT

```
'THEN' 'GOTO' EE 'END'..
'END'..
'IF' K 'EQUAL' M 'THEN' 'BEGIN'SOM1.=T.. SOM2.=DNT..SOM3.=DTT..
SOM4.=DTDNT 'END'.. 'IF' K 'GREATER' M 'THEN'
'BEGIN' SOM1.=SOM1+T.. SOM3.=SOM3+DTT.. SOM2.=SOM2+DNT..
SOM4.=SOM4+DTDNT..
'IF' ABS(SOM1*SOM2*SOM3*SOM4) 'LESS' '-300 'THEN' 'GOTO' DD..
'IF' ABS(T/SOM1) 'GREATER' TOL 'OR' ABS(DNT/SOM2) 'GREATER' TOL 'OR'
ABS(DTT/SOM3) 'GREATER' TOL 'OR' ABS(DTDNT/SOM4) 'GREATER' TOL
'THEN' 'GOTO' DD 'END' 'ELSE' 'GOTO' DD..
EE..
PSIT(/M,3,0/).=PSI.=TAU'POWER'(-.5*M)*(.5*LN(TAU)*SOM1+ST+SOM2)..
PSIT(/M,3,1/).=DTPSI.=-.5*M/TAU*PSI+TAU'POWER'(-.5*M)*
(.5/TAU*SOM1+SDTT+SOM4+.5*LN(TAU)*SUM3) 'END'..
'END' M CYCLE..
'END'..

'REAL' 'PROCEDURE' EPSALG(N,P,INF,TN,TOL,NRES)..
       'VALUE' P,INF,TOL.. 'REAL' TN,TOL.. 'INTEGER' N,P,INF,NRES..
'BEGIN' 'REAL' AUX0,AUX1,AUX2,RES,TOLR,TOLH..
        'INTEGER' M,S,I,E,MMAX..
        'ARRAY' L(/0..INF-P/),EPS(/-1..1/)..
        MMAX.=INF-P.. M.=I.=1..
        N.=P.. AUX0.=TN..
        N.=P+1.. AUX1.=TN..L(/0/).=AUX0+AUX1..L(/1/).=1.0/(AUX1+'-200)..
NEW L.. I.=-I.. E.=(I-1)/2.. EPS(/-1/).=L(/M+E/).. M.=M+1..
        N.=N+1.. AUX1.=TN.. AUX0.=L(/0/)+AUX1.. AUX1.=1.0/(AUX1+'-200)..
        'FOR' S.=2 'STEP' 1 'UNTIL' M 'DO'
        'BEGIN' AUX2.=L(/S-2/)+1.0/(AUX1-L(/S-1/)+'-200)..
                L(/S-2/).=AUX0.. AUX0.=AUX1.. AUX1.=AUX2
        'END'..
        L(/M-1/).=AUX0.. L(/M/).=AUX1..
        'IF' M 'LESS' 3.5 'THEN' 'GOTO' NEW L.. RES.=L(/M-1-E/)..
        TOLR.=ABS(RES-EPS(/-I/)).. TOLH.=ABS(RES-EPS(/I/))..
        'IF' (TOLR 'GREATER' TOL 'OR' TOLH 'GREATER' TOL)
        'AND' M 'LESS' MMAX 'THEN' 'GOTO' NEW L..
        NRES.=N.. EPSALG.=
         'IF'RES'LESS'EPS(/ 1/)'AND'EPS(/ 1/)'LESS'EPS(/-1/)'THEN'
         EPS(/ 1/) 'ELSE'
         'IF'RES'LESS'EPS(/-1/)'AND'EPS(/-1/)'LESS'EPS(/ 1/)'THEN'
         EPS(/-1/) 'ELSE' RES
'END' EPSALG..

'REAL' 'PROCEDURE' BR(D).. 'INTEGER' D..
'BEGIN' 'REAL' FN,CN,FNL,EPSON.. 'INTEGER' U,V,C1,C2..
'ARRAY' L,HLP,EPSN(/1..2/), FT(/0..1/)..
   'IF' K'GREATER'0'THEN' 'BEGIN' U.=FNR/10.. V.=1+FNR/15..
FN.=(N+'-20)'POWER'U/(N+(-1)'POWER'FNR+('IF'FNR'LESS'3'THEN'.5
'ELSE' 0)) 'POWER' V 'END' 'ELSE' FN.=1..
CN.='IF' INT'LESS' 5 'THEN' 1 'ELSE' C(/N/)..
'IF' II 'LESS' 3 'THEN' 'BEGIN' 'FOR' RI.=R,I 'DO'
L(/RI/).=LARRAY(/II,LNR,N,RI/) 'END' 'ELSE'
'BEGIN' 'IF'(-1)'POWER'LNR'GREATER' 0 'THEN''BEGIN'L(/2/).=0..
L(/1/).='IF'LNR'EQUAL'20'THEN'1+1/N'ELSE'-1/(N*N)'END' 'ELSE'
'BEGIN' C1.=ENTIER(S/10+.01).. C2.=S-10*C1..
FNL.=1-1/N.. EPSON.=EPSZERO'POWER'N..
EPSN(/1/).=EPSON*COS(2*N*ALFA).. EPSN(/2/).=EPSON*SIN(2*N*ALFA)..
'IF'C1'EQUAL'3'THEN'
```

```
'BEGIN'HLP(/1/).=FNL*(EPSN(/1/)*LNEPS0-EPSN(/2/)*2*ALFA)..
    HLP(/2/).=FNL*(EPSN(/1/)*2*ALFA+EPSN(/2/)*LNEPS0)'END'..
'IF'C2'EQUAL'1'THEN'EPSN(/1/).=EPSN(/1/)-1..
'IF'C1'EQUAL'2'THEN''BEGIN'L(/1/).=FNL*EPSN(/1/)..
    L(/2/).=FNL*EPSN(/2/) 'END'
'ELSE' 'BEGIN' L(/1/).=HLP(/1/)+EPSN(/1/)/(N*N)..
L(/2/).=HLP(/2/)+EPSN(/2/)/(N*N) 'END'..
'IF' C2'EQUAL'1'THEN''BEGIN' HLP(/1/).=L(/1/)..
  L(/1/).=-L(/2/).. L(/2/).=HLP(/1/).. 'END'
'END'  'END'..
FT(/D/).=PSIT(/N,TNR,D/)..
'FOR'RI.=R,I'DO''BEGIN'L(/RI/).=SIGNL(/RI/)*L(/RI/)..
  LA(/CO(/RI/)/).=L(/RI/) 'END'..
'FOR'RI.=R,I'DO'
 B(/RI,D/).=FN*CN*FT(/D/)*PSITI(/N,TINR,0/)*L(/CO(/RI/)/)..
BR.= B(/R,D/) 'END'..

'REAL' 'PROCEDURE' P(TEST).. 'VALUE'TEST.. 'INTEGER'TEST..
'BEGIN' 'REAL' A,HLP.. 'INTEGER'J,P10..'INTEGER''ARRAY'POS(/1..8/)..
S.='IF' K 'GREATER' 0 'THEN' Q(/K/) 'ELSE'
Q(/QN1(/K/)/).. P10.=1000000000.. CO(/1/).='IF'S'LESS'0'THEN'2'ELSE'1..
CO(/2/).=3-CO(/1/)..  S.=ABS(S)..
'FOR'J.=1'STEP'1'UNTIL'8'DO'
'BEGIN'POS(/J/).=S'DIV'P10.. S.=S-P10*POS(/J/).. P10.=(P10+1)'DIV'10
'END'..
II.=POS(/1/).,TYPE.=POS(/2/).,N1.=POS(/3/).. INT.=POS(/4/)..
FNR.=POS(/5/).. TNR.=POS(/6/).,TINR.=POS(/7/).,SLRI.=POS(/8/)..
LNR.=S.. A.=(2-N1)/4..
SIGNL(/1/).='IF' SLRI 'LESS' 3 'THEN' 1 'ELSE' -1..
SIGNL(/2/).=(-1)'POWER'SLRI..
AA.. 'IF'K'LESS'0'THEN''BEGIN'N.=1..
    HLP.=FACT(/TYPE/)*FGAM(/II/).. 'IF' TEST'NOTEQUAL'2'THEN'
PHI.=HLP*(BR(0)*SIN(TH)+B(/I,0/)*COS(TH))..
'IF'TEST'EQUAL'4'THEN''BEGIN'DPDT(/K/).=HLP*(BR(1)*SIN(TH)+B(/I,1/)*
COS(TH)).,DPDTH(/K/).=HLP*(B(/R,0/)*COS(TH)-B(/I,0/)*SIN(TH)) 'END'
'END' 'ELSE' 'BEGIN'
HLP.=FGAM(/II/)*FACT(/TYPE/)..
'IF'TEST'NOTEQUAL'2'THEN'PHI.=
HLP*EPSALG(N,N1,MAX-1,BR(0)*SIN((N+A)*TH)+B(/I,0/)*COS((N+A)*TH),TOL,
LMAX(/1,K/))..
'IF'TEST'EQUAL'4'THEN''BEGIN'DPDT(/K/).=HLP*
EPSALG(N,N1,MAX-1,BR(1)*SIN((N+A)*TH)+B(/I,1/)*COS((N+A)*TH),TOL,
LMAX(/4,K/))..
DPDTH(/K/).=HLP*EPSALG(N,N1,MAX-1,
(N+A)*(BR(0)*COS((N+A)*TH)-B(/I,0/)*SIN((N+A)*TH)),TOL,
LMAX(/5,K/)) 'END'..
'IF'TYPE'EQUAL'3'THEN''BEGIN'HLP.=FGAM(/II/)..
'IF'TEST'EQUAL'4'THEN'DPDTH(/K/).=DPDTH(/K/)+PHI/TH 'END'..
'END' TEST SIGN K..
P.=PHI
'END' 'PROCEDURE' P..

'REAL' 'PROCEDURE' SOM(TK)..'REAL' TK..
'BEGIN' 'REAL' SM.. SM.=0.. 'FOR' K.=-14 'STEP' 1 'UNTIL' 48 'DO'
'BEGIN' LC.=LOC(/K/).. SM.=SM+('IF'ABS(LC)'LESS''-4
'THEN'0'ELSE'LC*TK) 'END'.. SOM.=SM 'END'..
```

```
    'PROCEDURE' NEWTON2(X,Y,H,K,F,G,PROCFG,EX,EY).,
    'VALUE' EX,EY., 'REAL' X,Y,H,K,F,G,EX,EY., 'PROCEDURE' PROCFG.,
    'BEGIN' 'COMMENT'.,
      'REAL' X0,Y0,H0,K0,FXY,GXY,FH,GH,DFDX,DGDX,DFDY,DGDY,DET,ABSH,ABSK.,
      'INTEGER' IT., 'BOOLEAN' READY.,
      X0.=X., Y0.=Y., H0.=H., K0.=K., IT.=0., READY.='FALSE'.,
  ANEW..  IT.=IT+1., OUTPUT(61,'('/')').,
      PROCFG.,
      'IF' READY 'THEN'
      'BEGIN' OUTPUT(41,'(''('   LAST STEP')'')')., 'GOTO' ENDPROC 'END'.,
      OUTPUT(41,'(''('   STEP')',ZZD')',IT).,
      FXY.=F., GXY.=G.,
      X.=X0+H., PROCFG., FH.=F., GH.=G.,
      X.=X0-H., PROCFG.,
      DFDX.=.5*(FH-F)/H., DGDX.=.5*(GH-G)/H.,
      X.=X0.,
      Y.=Y0+K., PROCFG., FH.=F., GH.=G.,
      Y.=Y0-K., PROCFG.,
      DFDY.=.5*(FH-F)/K., DGDY.=.5*(GH-G)/K.,
      DET.=DFDX*DGDY-DGDX*DFDY.,
      H.=(GXY*DFDY-FXY*DGDY)/DET.,
      K.=(FXY*DGDX-GXY*DFDX)/DET.,
      ABSH.=ABS(H)., ABSK.=ABS(K).,
      'IF' EX 'LESS' ABSH 'OR' EY 'LESS' ABSK 'THEN'
      'BEGIN' DET.=SQRT((H*H+K*K)/(H0*H0+K0*K0)).,
        'IF' 1.0 'LESS' DET 'THEN' 'BEGIN' H.=H/DET., K.=K/DET 'END'.,
        X.=X0.=X0+H., Y.=Y0.=Y0+K.,
        'IF' ABS(H) 'LESS' EX 'THEN' H.=EX.,
        'IF' ABS(K) 'LESS' EY 'THEN' K.=EY.,
        'IF' 5.5 'LESS' IT 'THEN' READY.='TRUE'.,
      'END' 'ELSE'
      'BEGIN' X.=X0+H., Y.=Y0+K., READY.='TRUE''END'.,
      'GOTO' ANEW.,
      ENDPROC..
    'END' NEWTON2.,

  'PROCEDURE' PTPTH.,
  'BEGIN''IF' ABS(T-TV) 'GREATER' '-7 'THEN' 'BEGIN' CHAPLYGIN(T,PSIT).,
    TV.=T 'END'.,
    FACT(/3/).=TH., PHI2.=SOM(P(4)).,
    PHIT.=SOM(DPDT(/K/)).,
    PHITH.=SOM(DPDTH(/K/)).,
    OUTPUT(41,'('/,5(+D.9D'+ZZD88)')',T,-TH*RAD,PHI2,PHIT,-PHITH)
  'END'.,

  INARRAY(40,Q)., INARRAY(40,QN1)., INARRAY(40,LOC).,
  PI.=4*ARCTAN(1)., RAD.=180/PI.,  TOL.='-5.,
  INPUT(40,'('')',CASE,EPSZERO,ALFA,GAMMA).,
  OUTPUT(41,'(''('CASE')',8B,ZD,/,'('EPSILON(0)=')',D.6D,/,
  '('ALFA      =')',D.6D,/,'('GAMMA     =')',D.6D')',
  CASE,EPSZERO,ALFA,GAMMA).,

    'BEGIN' 'ARRAY' PARAM(/1..6/).,
    SKIPF(43).,
    EOF(43,ALARM)., 'GOTO' SEARCH.,
  ALARM..
    OUTPUT(41,'('//,'('CASE UNKNOWN ON TAPE')'')')., 'GOTO' EOP.,
  SEARCH..
```

```
   GETARRAY(43,PARAM).. GETARRAY(43,LARRAY)..
   'IF' ABS(CASE-PARAM(/3/))'GREATER'0.5 'OR' ABS(EPSZERO-PARAM(/4/))
   'GREATER' '-8 'OR' ABS(ALFA-PARAM(/5/)) 'GREATER' '-8 'OR' ABS(GAMMA-
   PARAM(/6/)) 'GREATER' '-8 'THEN' 'GOTO' SEARCH..
   'END'..
   REWIND(43)..
 R.=1.. I.=2..
 GAMMA.=.5*GAMMA/PI..
 FACT(/1/).=1.0.. FACT(/2/).=PI.. LNEPSO.=LN(EPSZERO)..
 FGAM(/1/).=SQRT(1-.25*GAMMA*GAMMA)..
 FGAM(/2/).=-.25*GAMMA*GAMMA/FGAM(/1/).. FGAM(/3/).=.5*GAMMA..
 C(/1/).=1.25.. 'FOR' N.=2 'STEP' 1 'UNTIL' 100 'DO'
 'BEGIN' C(/N/).=N-2.5+1.25*N*(N+1)-1.0.. 'FOR' K.=2 'STEP' 1
   'UNTIL' N 'DO' C(/N/).=C(/N/)*((N-2.5)/K+1.25*N*(N+1)/(K*K)-1.0)
 'END'..
 DTAU.=.005.. DTHETA.=.5/RAD..
 INPUT(40,'('')',TI,T,THC)..  TH.=-THC/RAD..  TV.=0..
 T.=T+'-9..
 OUTPUT(41,'('*,'('TAU 1 =')',D.6D')',TI)..
 OUTPUT(41,'('//,5B,'('TAU(C)')',12B,'('THETA(C)')',14B,
 '('PSI')',13B,'('DPSI/DTAU')',9B,'('DPSI/DTHETA')',/')')..
 'BEGIN' 'INTEGER' KT,KH..
   KT.=1200*TI..
 READ..
         GETARRAY(43,PSITI)..  KH.=600/PSITI(/0,2,1/)..
   'IF' KH 'NOTEQUAL' KT 'THEN' 'GOTO' READ..
 'END'..
 'FOR' N.=0 'STEP' 1 'UNTIL' 100 'DO' 'FOR' J.=1,2,3,4,5 'DO'
 PSITI(/N,J,0/).=PSITI(/N,J,0/)+2*TI*PSITI(/N,J,1/)..
 NEWTON2(TH,T,DTHETA,DTAU,PHIT,PHITH,PTPTH,'-5,'-5)..
 'END'..   EOP..  'END'
          'EOP'
```

```
        LINE    0           PROGRAM BEGINS        (MESSAGE)              1
        LINE  262           PROGRAM ENDS          (MESSAGE)              1
        LINE  262           SOURCE DECK ENDS      (MESSAGE)              1

                THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE       F.I.L.X



CORE MAP  14.00.46.  NORMAL              CONTROL                                                      000100     032254     030054     002200
        ---TIME---LOAD MODE --L1--L2-----TYPE---------------USER-------CALL-------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
        FWA LOADER  073741  FWA TABLES  071032
        -PROGRAM----ADDRESS-                          --LABELED---COMMON--
        XXALGOL     000340                                     DATA        000100
        ALGORUN     011340                                     DATA        000100

        ALGLB00     013712                                     DATA        000100
        ALGLB01     017127                                     DATA        000100
        ALGLB02     017656                                     DATA        000100
        ALGLB03     025525                                     DATA        000100
        ALGLB05     026360                                     DATA        000100
        ALGLB06     027152                                     DATA        000100
        --ENTRY-----ADDRESS-                                            REFERENCES
        XXALGOL     010375
        ALGORUN     011340             XXALGOL

        ALGLB00     013712             XXALGOL

        ALGLB01     017127             XXALGOL

        ALGLB02     017656             XXALGOL

        ALGLB03     025530             XXALGOL

        ALGLB05     026360             XXALGOL

        ALGLB06     027152             XXALGOL

        ----UNSATISFIED EXTERNALS-----                                  REFERENCES


CHANNEL,60=INPUT,P80,R
CHANNEL,61=OUTPUT,P136,PP60,R
CHANNEL,40=60
CHANNEL,41=61
CHANNEL,43=LU43,A,B
CHANNEL,END
```

```
≠BEGIN≠ ≠COMMENT≠ THE COMPUTATION OF A QUASI-ELLIPTICIL AEROFOIL
                  IN A CIRCULATORY TRANSONIC POTENTIAL FLOW
                  BY USING LIGHTHILL S SECOND INTEGRAL OPERATOR..
≠INTEGER≠ MAX,CASE..
MAX.=100..
≠BEGIN≠ ≠REAL≠ EPSZERO,ALFA,GAMMA,TZ1,TI,T,TH,LNEPS0,TI1,T1,SQT1,SQTI1,
PI,SI,MACH,CP,RR,RAD,DXDTH,DYDTH,EXPSI,
MU2,L43,L45,
INTEG,QTI,QT,FT,PHI,TOL,ZERO,Z1ABS, LC,  X1,X2,Y1,Y2,TH1,TH2,PHI1,PHI2,
ABSTH,X,Y,PHIT,PHITH,JP,PD2,TP,FTN1,FTN2,X3,Y3..
≠INTEGER≠ MAXIT,RECO,K,II,N,J,D,TYPE,NI,INT,FNP,TINR,SLRI,LNR,R,I,RI,S,
ATAU,LOWER,UPPER,SIG,KEN,TEL,TT,AK,TK,SIDE,ATH,TTH,TNR,H,XY,
CUS,ISOTA,TAUCUS..
≠BOOLEAN≠ CUSP,ISOTAU,TAUCUSP,MU2IN..
≠ARRAY≠ FACT,FGAM(/1..3/),Q(/1..48/),LARRAY(/1..2,1..11,0..100,1..2/),
CE(/3..5,0..MAX,1..2/),RL(/1..3/),MULT(/1..4/),DUM(/1..2/),
XE,YE,PHITE,PHITHE(/1..12/),PSIT(/0..140,1..5,0..1/),C(/1..MAX/),
PSITI(/0..140,1..5,0..0/),
B(/1..2,0..1/),PSITN1(/0..1/),XP,YP,DPOT,DPDTH(/-14..48/),
LOC(/1..3,1..2,0..1,-14..48/),DX,DY(/1..3,1..2,0..1/),
AB,BB(/2..3/),
CS(/1..4/),LA,XTI,XPI,YTI,YPI,Z1(/1..2/)..
≠INTEGER≠ ≠ARRAY≠ CONV(/1..3/),CO,SIGNL(/1..2/),QN1(/-14..-1/),
ELOC(/1..3,1..2,0..1,1..12/),LMAX(/1..5,1..48/),ELMAX(/1..5,3..10/)..

≠PROCEDURE≠ CHAPLYGIN(TAU,PSIT).. ≠REAL≠ TAU.. ≠ARRAY≠ PSIT..
≠BEGIN≠ ≠COMMENT≠ CHAPLYGINFUNCTIONS FOR TAU LESS .05 ..
≠REAL≠ TOL,    H,T,DTT,DNT,DTDNT,SOM1,SOM2,SOM3,SOM4,F1K,F2K,
PSI,DTPSI,DNPSI,DTDNPSI,ST,SDTT..
≠INTEGER≠ M,K,A,B,I,D..
≠REAL≠≠PROCEDURE≠N..N.=A*M+.5*B..
    TOL.=≠-11..
    PSIT(/0,3,0/).=PSIT(/0,3,1/).=0..
PSIT(/1,3,0/).=TAU≠POWER≠(-.5).. PSIT(/1,3,1/).=-.5*TAU≠POWER≠(-1.5)..
≠FOR≠ M.=0 ≠STEP≠ 1 ≠UNTIL≠ 100 ≠DO≠
≠BEGIN≠ A.=1.. B.=0..
AA.. T.=SOM1.=1.. I.=(-7*A+6*B+9)/2..
DTT.=SOM2.=DNT.=SOM3.=DTDNT.=SOM4.=0..
≠IF≠M≠EQUAL≠0≠AND≠B≠EQUAL≠0≠THEN≠≠GOTO≠CC.. K.=0..
BB.. K.=K+1.. F1K.=(-1.25*N*(N+1)+(K-1)*(N+K-3.5))/(N+K)/K*TAU..
≠IF≠ B≠EQUAL≠ 0 ≠THEN≠
≠BEGIN≠F2K.=1.25/K*(-1+(K-1)/(N+K)≠POWER≠2*(K+2.8))*TAU..
DTDNT.=F1K*(DTDNT+DNT/TAU)+F2K*(DTT+T/TAU)..
DNT.=F1K*DNT+F2K*T.. SOM3.=SOM3+DNT.. SOM4.=SOM4+DTDNT ≠END≠..
DTT.=F1K*(DTT+T/TAU).. T.=F1K*T.. SOM1.=SOM1+T.. SOM2.=SOM2+DTT..
≠IF≠ ABS(T/SOM1) ≠GREATER≠ TOL ≠OR≠ ABS(DTT/SOM2) ≠GREATER≠ TOL
≠THEN≠ ≠GOTO≠ BB..  ≠IF≠ B ≠NOTEQUAL≠ 0 ≠THEN≠ ≠GOTO≠ CC..
≠IF≠ ABS(DNT/SOM3) ≠GREATER≠ TOL ≠OR≠ ABS(DTDNT/SOM4) ≠GREATER≠ TOL
≠THEN≠ ≠GOTO≠ BB..
CC..H.=TAU≠POWER≠(.5*N).. PSIT(/M,I,0/).=PSI.=H*SOM1..
PSIT(/M,I,1/).=DTPSI.=.5*N/TAU*PSI+H*SOM2.. ≠IF≠ B≠EQUAL≠ 0 ≠THEN≠
≠BEGIN≠ PSIT(/M,2,0/).=DNPSI.=.5*LN(TAU)*PSI+H*SOM3..
PSIT(/M,2,1/).=DTDNPSI.=.5*LN(TAU)*DTPSI+.5/TAU*PSI+H*(.5*N/TAU*SOM3+
  SOM4)..
B.=1.. ≠GOTO≠ AA ≠END≠..
≠IF≠ B≠EQUAL≠1 ≠THEN≠ ≠BEGIN≠ A.=-1.. B.=-1.. ≠GOTO≠ AA ≠END≠..
≠IF≠ M ≠GREATER≠ 1 ≠THEN≠
≠BEGIN≠ K.=0.. T.=ST.=1.. DTT.=SDTT.=DNT.=DTDNT.=0..
```

APPENDIX D

LISTING OF AIRFOIL

```
SOM1.=SOM2.=SOM3.=SOM4.=0..
DD.. K.=K+1.. ≠IF≠ K ≠NOTEQUAL≠ M ≠THEN≠
≠BEGIN≠ F1K.=(-1.25*M*(M-1)+(K-1)*(K-M-3.5))/(K-M)/K*TAU.,
F2K.=1.25/K*(-1+(K-1)/(K-M)/(K-M)*(K+2.8))*TAU ≠END≠ ≠ELSE≠
≠BEGIN≠F1K.=1.25*(1-M)/M*(M+2.8)*TAU.,
  F2K.=(3.5-2.25/M)*TAU ≠END≠..
DTDNT.=F1K*(DTDNT+DNT/TAU)+F2K*(DTT+T/TAU).. DNT.=F1K*DNT+F2K*T..
DTT.=F1K*(DTT+T/TAU).. T.=F1K*T..
≠IF≠ K≠LESS≠M≠THEN≠≠BEGIN≠ST.=ST+T.. SDTT.=SDTT+DTT..
≠IF≠ ABS(T/ST) ≠LESS≠ TOL  ≠AND≠ ABS(DTT/SDTT) ≠LESS≠ TOL
≠THEN≠ ≠GOTO≠ EE ≠END≠..
≠IF≠ K ≠EQUAL≠ M ≠THEN≠ ≠BEGIN≠SOM1.=T., SOM2.=DNT.,SOM3.=DTT.,
SOM4.=DTDNT ≠END≠.. ≠IF≠ K ≠GREATER≠ M ≠THEN≠
≠BEGIN≠ SOM1.=SOM1+T., SOM3.=SOM3+DTT., SOM2.=SOM2+DNT..
SOM4.=SOM4+DTDNT..
≠IF≠ ABS(T/SOM1) ≠GREATER≠ TOL ≠OR≠ ABS(DNT/SOM2) ≠GREATER≠ TOL ≠OR≠
ABS(DTT/SOM3) ≠GREATER≠ TOL ≠OR≠ ABS(DTDNT/SOM4) ≠GREATER≠ TOL
≠THEN≠ ≠GOTO≠ DD ≠END≠ ≠ELSE≠ ≠GOTO≠ DD..
EE..
PSIT(/M,3,0/).=PSI.=TAU≠POWER≠(-.5*M)*(.5*LN(TAU)*SOM1+ST+SOM2)..
PSIT(/M,3,1/).=DTPSI.=-.5*M/TAU*PSI+TAU*POWER≠(-.5*M)*
(.5/TAU*SOM1+SDTT+SOM4+.5*LN(TAU)*SOM3) ≠END≠..
≠END≠ M CYCLE..
≠END≠..

≠REAL≠ ≠PROCEDURE≠ CIS(KK).. ≠INTEGER≠ KK..
≠BEGIN≠ CS(/1/).=(-TH*COS((N-1)*TH)+SIN((N-1)*TH)/(N-1))/(N-1)..
CS(/2/).=(-TH*COS((N+1)*TH)+SIN((N+1)*TH)/(N+1))/(N+1)..
CS(/3/).=(TH*SIN((N-1)*TH)+COS((N-1)*TH)/(N-1))/(N-1)..
CS(/4/).=(TH*SIN((N+1)*TH)+COS((N+1)*TH)/(N+1))/(N+1)..
CIS.=CS(/KK/) ≠END≠..

≠REAL≠ ≠PROCEDURE≠ EPSALG(N,P,INF,TN,TOL,NRES)..
        ≠VALUE≠ P,INF,TOL.. ≠REAL≠ TN,TOL.. ≠INTEGER≠ N,P,INF,NRES..
≠BEGIN≠ ≠REAL≠ AUX0,AUX1,AUX2,RES,TOLR,TOLH..
        ≠INTEGER≠ M,S,I,E,MMAX..
        ≠ARRAY≠ L(/0..INF-P/),EPS(/-1..1/)..
        MMAX.=INF-P.. M.=I.=1..
        N.=P.. AUX0.=TN..
        N.=P+1., AUX1.=TN.,L(/0/).=AUX0+AUX1.,L(/1/).=1.0/(AUX1+≠-200)..
NEW L.. I.=-I., E.=(I-1)/2.. EPS(/-I/).=L(/M+E/).. M.=M+1.,
        N.=N+1., AUX1.=TN., AUX0.=L(/0/)+AUX1., AUX1.=1.0/(AUX1+≠-200)..
        ≠FOR≠ S.=2 ≠STEP≠ 1 ≠UNTIL≠ M ≠DO≠
        ≠BEGIN≠ AUX2.=L(/S-2/)+1.0/(AUX1-L(/S-1/)+≠-200)..
                L(/S-2/).=AUX0.. AUX0.=AUX1.. AUX1.=AUX2
        ≠END≠..
        L(/M-1/).=AUX0.. L(/M/).=AUX1..
        ≠IF≠ M ≠LESS≠ 3.5 ≠THEN≠ ≠GOTO≠ NEW L.. RES.=L(/M-1-E/)..
        TOLR.=ABS(RES-EPS(/-I/)).. TOLH.=ABS(RES-EPS(/I/))..
        ≠IF≠ (TOLR ≠GREATER≠ TOL ≠OR≠ TOLH ≠GREATER≠ TOL)
        ≠AND≠ M ≠LESS≠ MMAX ≠THEN≠ ≠GOTO≠ NEW L..
        NRES.=N.. EPSALG.=
         ≠IF≠RES≠LESS≠EPS(/ 1/)≠AND≠EPS(/ 1/)≠LESS≠EPS(/-1/)≠THEN≠
         EPS(/ 1/) ≠ELSE≠
         ≠IF≠RES≠LESS≠EPS(/-1/)≠AND≠EPS(/-1/)≠LESS≠EPS(/ 1/)≠THEN≠
         EPS(/-1/) ≠ELSE≠ RES
≠END≠ EPSALG..
```

```
≠REAL≠ ≠PROCEDURE≠ BR(D).. ≠INTEGER≠ D..                                AIRFOIL  118
≠BEGIN≠ ≠REAL≠ FN.CN.FNL.EPSON.. ≠INTEGER≠ U.V.C1.C2..                  AIRFOIL  119
≠ARRAY≠ L.HLP.EPSN(/1..2/). FT(/0..1/)..                                AIRFOIL  120
   ≠IF≠ K≠GREATER≠0≠THEN≠ ≠BEGIN≠ U.=FNR/10.. V.=1+FNR/15..             AIRFOIL  121
FN.=(N+≠-20)≠POWER≠U/(N+(-1)≠POWER≠FNR+(≠IF≠FNR≠LESS≠3≠THEN≠.5          AIRFOIL  122
≠ELSE≠ 0)) ≠POWER≠ V ≠END≠ ≠ELSE≠ FN.=1..                               AIRFOIL  123
CN.=≠IF≠ INT≠LESS≠ 5 ≠THEN≠ 1 ≠ELSE≠ C(/N/)..                           AIRFOIL  124
≠IF≠ II ≠LESS≠ 3 ≠THEN≠ ≠BEGIN≠ ≠FOR≠ RI.=R.I ≠DO≠                      AIRFOIL  125
L(/RI/).=LARRAY(/II.LNR.N.RI/) ≠END≠ ≠ELSE≠                             AIRFOIL  126
≠BEGIN≠ ≠IF≠(-1)≠POWER≠LNR≠GREATER≠ 0 ≠THEN≠≠BEGIN≠L(/2/).=0..          AIRFOIL  127
L(/1/).=≠IF≠LNR≠EQUAL≠20≠THEN≠1+1/N≠ELSE≠-1/(N*N)≠END≠ ≠ELSE≠           AIRFOIL  128
≠BEGIN≠ C1.=ENTIER(S/10+.01).. C2.=S-10*C1..                            AIRFOIL  129
FNL.=1-1/N.. EPSON.=EPSZERO≠POWER≠N..                                   AIRFOIL  130
EPSN(/1/).=EPSON*COS(2*N*ALFA).. EPSN(/2/).=EPSON*SIN(2*N*ALFA)..       AIRFOIL  131
≠IF≠C1≠EQUAL≠3≠THEN≠                                                    AIRFOIL  132
≠BEGIN≠HLP(/1/).=FNL*(EPSN(/1/)*LNEPS0-EPSN(/2/)*2*ALFA)..              AIRFOIL  133
    HLP(/2/).=FNL*(EPSN(/1/)*2*ALFA+EPSN(/2/)*LNEPS0)≠END≠..            AIRFOIL  134
≠IF≠C2≠EQUAL≠1≠THEN≠EPSN(/1/).=EPSN(/1/)-1..                            AIRFOIL  135
≠IF≠C1≠EQUAL≠2≠THEN≠≠BEGIN≠L(/1/).=FNL*EPSN(/1/)..                      AIRFOIL  136
L(/2/).=FNL*EPSN(/2/) ≠END≠                                             AIRFOIL  137
≠ELSE≠ ≠BEGIN≠ L(/1/).=HLP(/1/)+EPSN(/1/)/(N*N)..                       AIRFOIL  138
L(/2/).=HLP(/2/)+EPSN(/2/)/(N*N) ≠END≠..                                AIRFOIL  139
≠IF≠ C2≠EQUAL≠1≠THEN≠≠BEGIN≠ HLP(/1/).=L(/1/)..                         AIRFOIL  140
   L(/1/).=-L(/2/).. L(/2/).=HLP(/1/).. ≠END≠                           AIRFOIL  141
≠END≠  ≠END≠..                                                          AIRFOIL  142
≠IF≠K≠NOTEQUAL≠-14≠THEN≠FT(/D/).=PSIT(/N.TNR.D/)                        AIRFOIL  143
≠ELSE≠ FT(/D/).=PSITN1(/D/)/2..                                         AIRFOIL  144
≠FOR≠RI.=R.I≠DO≠≠BEGIN≠L(/RI/).=SIGNL(/RI/)*L(/RI/)..                   AIRFOIL  145
 LA(/CO(/RI/)/).=L(/RI/) ≠END≠..                                        AIRFOIL  146
≠FOR≠RI.=R.I≠DO≠                                                        AIRFOIL  147
B(/RI.D/).=FN*CN*FT(/D/)*PSITI(/N.TINR.0/)*L(/CO(/RI/)/)..              AIRFOIL  148
BR.= B(/R.D/) ≠END≠..                                                   AIRFOIL  149
                                                                        AIRFOIL  150
≠REAL≠ ≠PROCEDURE≠ P(TEST).. ≠VALUE≠TEST.. ≠INTEGER≠TEST..              AIRFOIL  151
≠BEGIN≠ ≠REAL≠ A.HLP.. ≠INTEGER≠J.P10..≠INTEGER≠≠ARRAY≠POS(/1..8/)..    AIRFOIL  152
≠IF≠K≠EQUAL≠0≠THEN≠≠GOTO≠ AA.. S.=≠IF≠K≠GREATER≠0≠THEN≠Q(/K/) ≠ELSE≠    AIRFOIL  153
Q(/QN1(/K/)/).. P10.=1000000000.. CO(/1/).=≠IF≠S≠LESS≠0≠THEN≠2≠ELSE≠1.. AIRFOIL  154
CO(/2/).=3-CO(/1/)..  S.=ABS(S)..                                       AIRFOIL  155
≠FOR≠J.=1≠STEP≠1≠UNTIL≠8≠DO≠                                            AIRFOIL  156
≠BEGIN≠POS(/J/).=S≠DIV≠P10.. S.=S-P10*POS(/J/).. P10.=(P10+1)≠DIV≠10    AIRFOIL  157
≠END≠..                                                                 AIRFOIL  158
II.=POS(/1/)..TYPE.=POS(/2/)..N1.=POS(/3/).. INT.=POS(/4/)..            AIRFOIL  159
FNR.=POS(/5/).. TNR.=POS(/6/)..TINR.=POS(/7/)..SLRI.=POS(/8/)..         AIRFOIL  160
LNR.=S.. A.=(2-N1)/4..                                                  AIRFOIL  161
SIGNL(/1/).=≠IF≠ SLRI ≠LESS≠ 3 ≠THEN≠ 1 ≠ELSE≠ -1..                     AIRFOIL  162
SIGNL(/2/).=(-1)≠POWER≠SLRI..                                           AIRFOIL  163
AA.. ≠IF≠K≠LESS≠0≠THEN≠≠BEGIN≠N.=1..                                    AIRFOIL  164
    HLP.=FACT(/TYPE/)*FGAM(/II/).. ≠IF≠ TEST≠NOTEQUAL≠2≠THEN≠           AIRFOIL  165
PHI.=HLP*(BR(0)*SIN(TH)+B(/I.0/)*COS(TH))..                             AIRFOIL  166
≠IF≠TEST≠NOTLESS≠2≠THEN≠ ≠BEGIN≠ TP.=TH..                               AIRFOIL  167
XP(/K/).=HLP*FT*.5*(-.5*COS(2*TH)*(BR(1)-.5*BR(0)/T)+                   AIRFOIL  168
TH*(B(/I.1/)+.5*B(/I.0/)/T)+.5*SIN(2*TH)*(B(/I.1/)-.5*                  AIRFOIL  169
B(/I.0/)/T))-HLP*FTN1*LA(/R/)..                                         AIRFOIL  170
YP(/K/).=+HLP*(≠IF≠K≠EQUAL≠-14≠THEN≠FTN2*.5 ≠ELSE≠ FTN1) *              AIRFOIL  171
LA(/I/)+HLP*FT*.5*(-.5*COS(2*TH)*(B(/I.1/)-.5*B(/I.0/)/T)+TH*           AIRFOIL  172
(B(/R.1/)+.5*B(/R.0/)/T)-.5*SIN(2*TH)*(B(/R.1/)-.5*B(/R.0/)/T)) ≠END≠.. AIRFOIL  173
≠IF≠TEST≠EQUAL≠4≠THEN≠≠BEGIN≠DPDT(/K/).=HLP*(B(/R.1/)*SIN(TH)+B(/I.1/)* AIRFOIL  174
COS(TH))..DPDTH(/K/).=HLP*(B(/R.0/)*COS(TH)-B(/I.0/)*SIN(TH)) ≠END≠     AIRFOIL  175
```

```
#END# #ELSE# #BEGIN# #IF# K #EQUAL# 0 #THEN#
#BEGIN# TI1.=1-TI.. T1.=1-T.. SQT1.=SQRT(T1).. SQTI1.=SQRT(TI1)..
INTEG.=SQT1*(T1*(.2*T1+1/3)+1)+.5*LN(ABS((SQT1-1)/(SQT1+1)))-
(SQTI1*(TI1*(.2*TI1+1/3)+1)+.5*LN(ABS((SQTI1-1)/(SQTI1+1))))..
QTI.=2*TI*TI1#POWER#(-2.5).. QT.=2*T*T1#POWER#(-2.5)..
PHI.=-FGAM(/3/)*(INTEG-2*TI/QTI)..
XP(/K/).=-FT*FGAM(/3/)*SIN(TH)/QT..
YP(/K/).=FT*FGAM(/3/)*COS(TH)/QT..
DPDT(/K/).=-FGAM(/3/)/QT.. DPDTH(/K/).=0 #END#
#ELSE# #BEGIN# HLP.=FGAM(/II/)*FACT(/TYPE/)..
#IF#TEST#NOTEQUAL#2#THEN#PHI.=
HLP*EPSALG(N.N1.MAX-1.BR(0)*SIN((N+A)*TH)+B(/I.0/)*COS((N+A)*TH).TOL.
LMAX(/1.K/))..
#IF#TEST#EQUAL#4#THEN##BEGIN#DPDT(/K/).=HLP*
EPSALG(N.N1.MAX-1.BR(1)*SIN((N+A)*TH)+B(/I.1/)*COS((N+A)*TH).TOL.
LMAX(/4.K/))..
DPDTH(/K/).=HLP*EPSALG(N.N1.MAX-1.
(N+A)*(BR(0)*COS((N+A)*TH)-B(/I.0/)*SIN((N+A)*TH)).TOL.
LMAX(/5.K/)) #END#..
#IF#TYPE#EQUAL#3#THEN##BEGIN#HLP.=FGAM(/II/)..
#IF#TEST#EQUAL#4#THEN#DPDTH(/K/).=DPDTH(/K/)+PHI/TH #END#..
#IF# TEST #NOTLESS# 2 #THEN# #BEGIN# #IF# TYPE #NOTGREATER# 2 #THEN#
#BEGIN#XP(/K/).=.5*FT*HLP*EPSALG(N.N1.MAX-1.COS((N-1+A)*TH)/(N-1+A)*
(-BR(1)-.5*(N+A)/T*BR(0))+COS((N+1+A)*TH)/(N+1+A)*(-B(/R.1/)+.5*(N+A)
/T*B(/R.0/))+SIN((N-1+A)*TH)/(N-1+A)*(B(/I.1/)+.5*(N+A)/T*B(/I.0/))+
SIN((N+1+A)*TH)/(N+1+A)*(B(/I.1/)-.5*(N+A)/T*B(/I.0/)).TOL.
LMAX(/2.K/))..
YP(/K/).=.5*FT*HLP*EPSALG(N.N1.MAX-1.
SIN((N-1+A)*TH)/(N-1+A)*(BR(1)+.5*(N+A)/T* BR(0))+
SIN((N+1+A)*TH)/(N+1+A)*(-B(/R.1/)+.5*(N+A)/T*B(/R.0/))+
COS((N-1+A)*TH)/(N-1+A)*(B(/I.1/)+.5*(N+A)/T*B(/I.0/))+
COS((N+1+A)*TH)/(N+1+A)*(-B(/I.1/)+.5*(N+A)/T*B(/I.0/)).TOL.
LMAX(/3.K/)) #END#
#ELSE# #BEGIN# XP(/K/).=.5*FT*HLP*EPSALG(N.2.MAX-1.
    CIS(1)*(BR(1)+.5*N*BR(0)/T)+
    CS(/2/)*(B(/R.1/)-.5*N*B(/R.0/)/T)+
    CS(/3/)*(B(/I.1/)+.5*N*B(/I.0/)/T)+
    CS(/4/)*(B(/I.1/)-.5*N*B(/I.0/)/T)-
    .5* B(/R.0/)/T*(SIN((N-1)*TH)/(N-1)-SIN((N+1)*TH)/(N+1))-
    .5*B(/I.0/)/T*(COS((N-1)*TH)/(N-1)-COS((N+1)*TH)/(N+1)).TOL.
LMAX(/2.K/))..
YP(/K/).=.5*FT*HLP*EPSALG(N.2.MAX-1.
    CIS(3)*(BR(1)+.5*N*BR(0)/T)-
    CS(/4/)*(B(/R.1/)-.5*N*B(/R.0/)/T)-
    CS(/1/)*(B(/I.1/)+.5*N*B(/I.0/)/T)+
    CS(/2/)*(B(/I.1/)-.5*N*B(/I.0/)/T)-
    .5*B(/R.0/)/T*(COS((N-1)*TH)/(N-1)+COS((N+1)*TH)/(N+1))+
.5*B(/I.0/)/T*(SIN((N-1)*TH)/(N-1)+SIN((N+1)*TH)/(N+1)).TOL.
LMAX(/3.K/)) #END#
#END# #END# TEST K EQUAL 0.. #END# TEST SIGN K..
P.=#IF#TEST#EQUAL#2#THEN#XP(/K/)#ELSE#PHI#END#PROCEDURE P..

#REAL# #PROCEDURE# SOM(J.SIDE.TK)..#REAL#TK..#INTEGER# J.SIDE..
#BEGIN# #REAL# SM.. SM.=0.. #FOR# K.=-14 #STEP# 1 #UNTIL# 48 #DO#
#BEGIN# LC.=LOC(/J.SIDE.SIG.K/).. SM.=SM+(#IF#ABS(LC)#LESS##-4
#THEN# 0 #ELSE# LC*TK)..
#END#..
SOM.=SM
```

```
≠END≠ SOM.,

≠PROCEDURE≠ FTCONV.,
≠BEGIN≠FT.=SQRT(TI/T)*2*T/(1-T)≠POWER≠2.5.,
≠FOR≠D.=0,1≠DO≠PSITN1(/D/).=
(1-1.5*D)*T≠POWER≠(-.5-D)+1.25*PSIT(/1,1,D/).,
FTN1.=2.5*TI*SQRT(T)*((1-TI)/(1-T))≠POWER≠2.5*(PSIT(/1,2,1/)+.5/T*
(PSIT(/1,1,0/)+PSIT(/1,2,0/))).,
FTN2.=3.8*FTN1-6*TI*(1-TI)≠POWER≠2.5*LN(T).,
J.=≠IF≠ T ≠NOTGREATER≠ TZ1 ≠THEN≠ 1 ≠ELSE≠≠IF≠ T ≠NOTGREATER≠ TI ≠THEN≠
  2 ≠ELSE≠ 3.,
≠END≠ FTCONV.,

≠PROCEDURE≠ COMPC.,
≠BEGIN≠ ≠REAL≠ FN,V6,VTI.,
≠ARRAY≠ C1,C1ST,TLNZ1,TERM1,TERM2,TERM3(/1..2/).,
≠PROCEDURE≠COMUDI(A,B,C,T)., ≠INTEGER≠ T., ≠ARRAY≠ A,B,C.,
≠BEGIN≠≠REAL≠MOD.,≠ARRAY≠P,Q(/1..2/).,
P(/1/).=A(/1/)., P(/2/).=A(/2/)., Q(/1/).=B(/1/)., Q(/2/).=B(/2/).,
MOD.=≠IF≠T≠GREATER≠0≠THEN≠1≠ELSE≠1/(Q(/1/)*Q(/1/)+Q(/2/)*Q(/2/)).,
C(/1/).=(P(/1/)*Q(/1/)-T*P(/2/)*Q(/2/))*MOD.,
C(/2/).=(T*P(/1/)*Q(/2/)+P(/2/)*Q(/1/))*MOD.,
≠END≠.,
≠PROCEDURE≠ COPOWER(A,P,B)., ≠VALUE≠P.,≠REAL≠P.,≠ARRAY≠A,B.,
≠BEGIN≠≠REAL≠MOD,ARC.,MOD.=(A(/1/)*A(/1/)+A(/2/)*A(/2/))≠POWER≠(.5*P).,
ARC.=ARCTAN(A(/2/)/A(/1/)).,
ARC.=≠IF≠A(/1/)≠LESS≠0≠THEN≠P*(ARC+PI)≠ELSE≠P*ARC.,
B(/1/).=MOD*COS(ARC)., B(/2/).=MOD*SIN(ARC)., ≠END≠.,
V6.=SQRT(6)., SI.=SQRT((1-6*TI)/(1-TI)).,  VTI.=SQRT(TI).,
SI.=-.5*V6*LN(1.4+.4*V6)+.5*LN(.8)+.5*V6*LN((V6+SI)/(V6-SI))
-.5*LN((1+SI)/(1-SI)).,
MULT(/1/).=MULT(/2/).=MULT(/3/).=MULT(/4/).=1.,
INPUT(40,≠(≠≠)≠,T,TH).,
SIG.=0.5*(1+SIGN(TH)).,  SIDE.=2-SIG.,  TH.=-TH.,
INPUT(40,≠(≠≠)≠,MU2).,≠IF≠ABS(MU2)≠LESS≠ ≠-6≠THEN≠
≠BEGIN≠ MU2IN.=≠FALSE≠., MU2.=BB(/2/)/AB(/2/) ≠END≠≠ELSE≠MU2IN.=≠TRUE≠.,
TH.=TH/RAD., FACT(/3/).=TH.,
COPOWER(Z1,+.5,LA).,CE(/3,0,1/).=-LA(/1/).,CE(/3,0,2/).=LA(/2/).,
≠FOR≠N.=1≠STEP≠1≠UNTIL≠MAX≠DO≠≠BEGIN≠COMUDI(LA,Z1,LA,-1).,
FN.=(N-1.5)/N.,LA(/1/).=FN*LA(/1/).,CE(/3,N,1/).=-LA(/1/).,
CE(/3,N,2/).=LA(/2/).=FN*LA(/2/) ≠END≠.,
C1(/1/).=-CE(/3,1,1/)., C1(/2/).=CE(/3,1,2/).,LA(/1/).=0.,LA(/2/).=1.,
≠FOR≠N.=0≠STEP≠1≠UNTIL≠MAX≠DO≠≠BEGIN≠
COMUDI(LA,Z1,LA,1).,FN.=(N-.5)/(N+1).,CE(/4,N,1/).=LA(/1/).=FN*LA(/1/).,
CE(/4,N,2/).=LA(/2/).=FN*LA(/2/) ≠END≠.,
COPOWER(Z1,1.5,LA).,LA(/1/).=2/3*LA(/1/).,LA(/2/).=2/3*LA(/2/).,
C1ST(/1/).=LA(/1/)., C1ST(/2/).=LA(/2/).,
≠FOR≠N.=2≠STEP≠1≠UNTIL≠MAX≠DO≠≠BEGIN≠COMUDI(LA,Z1,LA,1).,
FN.=(N-1)/(N+.5)., LA(/1/).=FN*LA(/1/)., LA(/2/).=FN*LA(/2/).,
CE(/5,N,1/).=-LA(/1/)*N*C(/N/)., CE(/5,N,2/).=-LA(/2/)*N*C(/N/) ≠END≠.,
TLNZ1(/1/).=1+SI+LN(4*Z1ABS).,
TLNZ1(/2/).=ARCTAN(Z1(/2/)/Z1(/1/)).,
COMUDI(C1,TLNZ1,TLNZ1,1)., EXPSI.=EXP(SI)., ≠FOR≠RI.=1,2≠DO≠
≠BEGIN≠TERM1(/RI/).=VTI/EXPSI*TLNZ1(/RI/).,
TERM2(/RI/).=VTI*EXPSI*1.25*C1ST(/RI/).,
TERM3(/RI/).=PI*C1(/RI/)/EXPSI*VTI ≠END≠.,
XTI(/2/).=TERM1(/1/)+TERM2(/1/)., XTI(/1/).=-TERM1(/2/)-TERM2(/2/).,
YTI(/2/).=TERM1(/2/)-TERM2(/2/).,YTI(/1/).=TERM1(/1/)-TERM2(/1/).,
```

```
XPI(/2/).=TERM3(/2/)., XPI(/1/).=-TERM3(/1/).,
YPI(/2/).=TERM3(/1/)., YPI(/1/).=TERM3(/2/).,
≠END≠ COMPC.,

≠REAL≠≠PROCEDURE≠ PE(TEST)., ≠VALUE≠ TEST., ≠INTEGER≠ TEST.,
≠BEGIN≠ ≠REAL≠ PHI,F., ≠INTEGER≠ TEL,CNR,TEK,FK,A,M1., ≠BOOLEAN≠ XY.,
≠REAL≠≠PROCEDURE≠ BR(D)., ≠INTEGER≠ D.,
≠BEGIN≠≠ARRAY≠C12(/1..2/).,C12(/1/).=0., C12(/2/).=1.,≠IF≠N≠EQUAL≠-1
≠THEN≠≠BEGIN≠B(/R,D/).=PSIT(/0,4,D/)*EXP(-.5*SI)*C12(/CO(/1/)/).,
B(/I,D/).=PSIT(/0,4,D/)*EXP(-.5*SI)*TEK*C12(/CO(/2/)/)≠END≠ ≠ELSE≠
≠BEGIN≠ B(/R,D/).=PSIT(/N,TNR,D/)*EXP((N*M1+F)*SI).,
B(/I,D/).=B(/R,D/)*TEK*CE(/CNR,N,CO(/2/)/).,
B(/R,D/).=B(/R,D/)*CE(/CNR,N,CO(/1/)/) ≠END≠.,
BR.=B(/R,D/)  ≠END≠.,
XY.=TEST≠GREATER≠1≠AND≠TEST≠LESS≠5., TEL.=5*(RI-1)., CNR.=K-TEL.,
CO(/1/).=3-RI., CO(/2/).=RI., TEK.=2*RI-3., ≠IF≠K≠EQUAL≠1+TEL≠THEN≠
≠BEGIN≠PHI.=TEK*CE(/3,0,CO(/2/)/).,
XE(/K/).=YE(/K/).=PHITE(/K/).=PHITHE(/K/).=0 ≠END≠.,
≠IF≠K≠EQUAL≠2+TEL≠THEN≠≠BEGIN≠≠FOR≠D.=0,1≠DO≠≠BEGIN≠
B(/R,D/).=CE(/3,1,CO(/1/)/)*PSIT(/1,1,D/)*EXP(-SI).,
B(/I,D/).=TEK*CE(/3,1,CO(/2/)/)*PSIT(/1,1,D/)*EXP(-SI) ≠END≠.,
PHI.=B(/R,0/)*SIN(TH)+B(/I,0/)*COS(TH)., ≠IF≠XY≠THEN≠≠BEGIN≠
XE(/K/).=.5*FT*(-.5*COS(2*TH)*(B(/R,1/)-.5*B(/R,0/)/T)+TH*(B(/I,1/)+
.5*B(/I,0/)/T)+.5*SIN(2*TH)*(B(/I,1/)-.5*B(/I,0/)/T))-FTN1/
(EXP(SI)*2.5*SQRT(TI)*(1-TI)≠POWER≠2.5)*CE(/3,1,CO(/1/)/).,
YE(/K/).=.5*FT*(-.5*COS(2*TH)*(B(/I,1/)-.5*B(/I,0/)/T)+TH*(B(/R,1/)+
.5*B(/R,0/)/T)-.5*SIN(2*TH)*(B(/R,1/)-.5*B(/R,0/)/T))+FTN1/
(EXP(SI)*2.5*SQRT(TI)*(1-TI)≠POWER≠2.5)*TEK*CE(/3,1,CO(/2/)/)
≠END≠., ≠IF≠TEST≠GREATER≠3≠THEN≠≠BEGIN≠PHITE(/K/).=
B(/R,1/)*SIN(TH)+B(/I,1/)*COS(TH).,
PHITHE(/K/).=B(/R,0/)*COS(TH)-B(/I,0/)*SIN(TH) ≠END≠≠END≠.,
≠IF≠K≠GREATER≠2+TEL≠AND≠K≠LESS≠6+TEL≠THEN≠≠BEGIN≠FK.=ABS(CNR-4).,
F.=(1-FK)/2.,A.=3*FK-1., M1.=CNR*(9-CNR)-19.,   TNR.=5-4*FK.,
≠IF≠TEST≠NOTEQUAL≠2≠THEN≠PHI.=EPSALG(N,A,MAX-1,BR(0)*SIN((N+F)*TH)+
B(/I,0/)*COS((N+F)*TH),TOL,ELMAX(/1,K/)).,  ≠IF≠ XY ≠THEN≠ ≠BEGIN≠
XE(/K/).=.5*FT*EPSALG(N,A,MAX-1,COS((N+F-1)*TH)/(N+F-1)*(-BR(1)-.5*
(N+F)/T*BR(0))+COS((N+F+1)*TH)/(N+F+1)*(-B(/R,1/)+.5*(N+F)/T*B(/R,0/))
+SIN((N+F-1)*TH)/(N+F-1)*(B(/I,1/)+.5*(N+F)/T*B(/I,0/))
+SIN((N+F+1)*TH)/(N+F+1)*(B(/I,1/)-.5*(N+F)/T*B(/I,0/)),TOL,
ELMAX(/2,K/)).,
YE(/K/).=.5*FT*EPSALG(N,A,MAX-1,SIN((N+F-1)*TH)/(N+F-1)*(BR(1)+.5*
(N+F)/T*BR(0))+SIN((N+F+1)*TH)/(N+F+1)*(-B(/R,1/)+.5*(N+F)/T*B(/R,0/))
+COS((N+F-1)*TH)/(N+F-1)*(B(/I,1/)+.5*(N+F)/T*B(/I,0/))+
COS((N+F+1)*TH)/(N+F+1)*(-B(/I,1/)+.5*(N+F)/T*B(/I,0/)),TOL,
ELMAX(/3,K/)) ≠END≠.,
≠IF≠TEST≠GREATER≠3≠THEN≠≠BEGIN≠PHITE(/K/).=EPSALG(N,A,MAX-1,BR(1)
*SIN((N+F)*TH)+B(/I,1/)*COS((N+F)*TH),TOL,ELMAX(/4,K/)).,
PHITHE(/K/).=EPSALG(N,A,MAX-1,(N+F)*(BR(0)*COS((N+F)*TH)-B(/I,0/)*
SIN((N+F)*TH)),TOL,ELMAX(/5,K/)) ≠END≠ ≠END≠.,
≠IF≠K≠EQUAL≠11≠THEN≠≠BEGIN≠≠FOR≠D.=0,1≠DO≠B(/R,D/).=-PSIT(/2,1,D/)*
EXP(-2*SI)., PHI.=B(/R,0/)*SIN(2*TH)., ≠IF≠ XY ≠THEN≠ ≠BEGIN≠
XE(/11/).=FT*(COS(TH)/2*(-B(/R,1/)-B(/R,0/)/T)+COS(3*TH)/6*
(-B(/R,1/)+B(/R,0/)/T)).,YE(/11/).=FT*(SIN(TH)/2*(B(/R,1/)+B(/R,0/)/T)+
SIN(3*TH)/6*(-B(/R,1/)+B(/R,0/)/T)) ≠END≠.,
≠IF≠TEST≠GREATER≠3≠THEN≠≠BEGIN≠PHITE(/11/).=B(/R,1/)*SIN(2*TH).,
PHITHE(/11/).=2*B(/R,0/)*COS(2*TH) ≠END≠ ≠END≠.,
≠IF≠K≠EQUAL≠12≠THEN≠≠BEGIN≠≠FOR≠D.=0,1≠DO≠
B(/R,D/).=PSIT(/2,1,D/)*EXP(-2*SI)., PHI.=B(/R,0/)*COS(2*TH).,
```

```
≠IF≠XY≠THEN≠≠BEGIN≠XE(/12/).=FT*(SIN(TH)/2*(B(/R,1/)+B(/R,0/)/T)+
SIN(3*TH)/6*(B(/R,1/)-B(/R,0/)/T)).,  YE(/12/).=FT*(COS(TH)/2*
(B(/R,1/)+B(/R,0/)/T)+COS(3*TH)/6*(-B(/R,1/)+B(/R,0/)/T)) ≠END≠.,
≠IF≠TEST≠GREATER≠3≠THEN≠≠BEGIN≠PHITE(/12/).=B(/R,1/)*COS(2*TH).,
PHITHE(/12/).=-2*B(/R,0/)*SIN(2*TH) ≠END≠ ≠END≠.,
PE.=≠IF≠TEST≠EQUAL≠2≠THEN≠XE(/K/)≠ELSE≠PHI ≠END≠.,

≠REAL≠ ≠PROCEDURE≠ ESOM(RI,TERM).,≠REAL≠ TERM., ≠INTEGER≠ RI.,
≠BEGIN≠ ≠REAL≠ SOM., ≠INTEGER≠ LOCEL,KB,KE., SOM.=0.,
KB.=5*RI-4., KE.=≠IF≠RI≠LESS≠3≠THEN≠KB+4≠ELSE≠11.,
≠IF≠RI≠EQUAL≠4≠THEN≠≠BEGIN≠KB.=12.,KE.=12 ≠END≠.,
≠FOR≠K.=KB≠STEP≠1≠UNTIL≠KE≠DO≠≠BEGIN≠LOCEL.=ELOC(/J,SIDE,SIG,K/).,
≠IF≠ LOCEL ≠NOTEQUAL≠0≠THEN≠ SOM.=SOM+MULT(/RI/)*LOCEL*TERM ≠END≠.,
ESOM.=SOM  ≠END≠.,

≠REAL≠ ≠PROCEDURE≠ CORR(TERM)., ≠REAL≠ TERM., ≠BEGIN≠ ≠REAL≠ RESULT.,
RESULT.=0., ≠IF≠CUSP≠THEN≠≠FOR≠RI.=1,2,3,4≠DO≠
RESULT.=RESULT+ESOM(RI,TERM)., CORR.=RESULT ≠END≠.,

≠PROCEDURE≠ TAPE(T,AR)., ≠VALUE≠ T., ≠REAL≠ T., ≠ARRAY≠ AR.,
≠BEGIN≠ ≠INTEGER≠ KT,KAR.,
  KT.=1200*T.,
READ.,  GETARRAY(43,AR)., KAR.=600/AR(/0,2,1/).,
  ≠IF≠ KAR ≠NOTEQUAL≠ KT ≠THEN≠ ≠GOTO≠ READ
≠END≠.,

INARRAY(40,Q)., INARRAY(40,QN1)., INARRAY(40,LOC).,
≠FOR≠ J.=1,2,3 ≠DO≠ ≠FOR≠ SIDE.=1,2 ≠DO≠ ≠FOR≠ S.=0,1 ≠DO≠
≠FOR≠ K.=-11,-10,-9,-6,-5,-4,-1,31,34,35,36,37,40,41,42,43,46,47,48 ≠DO≠
LOC(/J,SIDE,S,K/).=0.,
INARRAY(40,ELOC)., PI.=3.14159265359., RAD.=180/PI.,
INPUT(40,≠(≠≠)≠,CASE,EPSZERO,ALFA,GAMMA,Z1(/1/),Z1(/2/),L43,L45).,
Z1ABS.=SQRT(Z1(/1/)≠POWER≠2+Z1(/2/)≠POWER≠2).,
≠FOR≠ J.=1,2,3 ≠DO≠ ≠FOR≠ SIDE.=1,2 ≠DO≠ ≠FOR≠ S.=0,1 ≠DO≠
≠BEGIN≠
  LOC(/J,SIDE,S,43/).=LOC(/J,SIDE,S,43/)+L43.,
  LOC(/J,SIDE,S,45/).=LOC(/J,SIDE,S,45/)+L45.,
≠END≠.,

  ≠BEGIN≠ ≠ARRAY≠ PARAM(/1..6/).,
  SKIPF(43).,
  EOF(43,ALARM)., ≠GOTO≠ SEARCH.,
ALARM.,
  OUTPUT(4),≠(≠//,≠(≠CASE UNKNOWN ON TAPE≠)≠≠)≠)., ≠GOTO≠ EOP.,
SEARCH.,
  GETARRAY(43,PARAM)., GETARRAY(43,LARRAY).,
  ≠IF≠ ABS(CASE-PARAM(/3/))≠GREATER≠0.5 ≠OR≠ ABS(EPSZERO-PARAM(/4/))
  ≠GREATER≠ ≠-8 ≠OR≠ ABS(ALFA-PARAM(/5/)) ≠GREATER≠ ≠-8 ≠OR≠ ABS(GAMMA-
  PARAM(/6/)) ≠GREATER≠ ≠-8 ≠THEN≠ ≠GOTO≠ SEARCH.,
  ≠END≠.,
  REWIND(43)., RECO.=0., GAMMA.=GAMMA/(2*PI).,
FGAM(/1/).=SQRT(1-.25*GAMMA*GAMMA)., FGAM(/2/).=-.25*GAMMA*GAMMA/
FGAM(/1/)., FGAM(/3/).=.5*GAMMA., C(/1/).=1.25.,
≠FOR≠N.=2≠STEP≠1≠UNTIL≠MAX≠DO≠
≠BEGIN≠ C(/N/).=N-2.5+1.25*N*(N+1)-1., ≠FOR≠K.=2≠STEP≠1≠UNTIL≠N≠DO≠
C(/N/).=C(/N/)*((N-2.5)/K+1.25*N*(N+1)/(K*K)-1) ≠END≠.,
INPUT(40,≠(≠≠)≠,CUS,TOL,MAXIT)., CUSP.=CUS=1.,
```

```
INPUT(40,≠(≠≠)≠,TI).,
≠BEGIN≠ ≠COMMENT≠ BEPALING VAN TAU(ZETA1).,
≠REAL≠ V6,LNZ1,UTI,A1,A2,F1,F2.,
≠REAL≠ ≠PROCEDURE≠ U(TAU)., ≠VALUE≠ TAU., ≠REAL≠ TAU.,
≠BEGIN≠ ≠REAL≠ P,RES.,
V6.=SQRT(6)., P.=SQRT((1-6*TAU)/(1-TAU)).,
RES.=LN((1+P)/(1-P)).,  P.=P/V6., RES.= RES-V6*LN((1+P)/(1-P)).,
U.=.5*RES  ≠END≠.,
                        V6.=SQRT(6)., LNZ1.=LN(Z1ABS)., UTI.=U(TI).,
A1.=TI., F1.=-LNZ1.,  A2.=TZ1.=TI -.005.,
REPEAT.. F2.=UTI-LNZ1-U(TZ1).,
≠IF≠ ABS(A2-A1) ≠LESS≠ ≠-6 ≠THEN≠ ≠GOTO≠ UIT.,
TZ1.=(A1*F2-A2*F1)/(F2-F1).,
≠IF≠ ABS(F2/F1) ≠LESS≠ 1 ≠THEN≠ ≠BEGIN≠ A1.=A2., F1.=F2 ≠END≠.,
A2.=TZ1., ≠GOTO≠ REPEAT.,
UIT..  ≠END≠.,
≠FOR≠SIG.=0,1≠DO≠≠FOR≠J.=1,2,3≠DO≠≠FOR≠SIDE.=1,2≠DO≠
INPUT(40,≠(≠≠)≠,DX(/J,SIDE,SIG/),DY(/J,SIDE,SIG/)).,
OUTPUT(41,≠(≠*,46S,/,41S,/,41S,///≠)≠,
≠(≠THE COMPUTATION OF A QUASI-ELLIPTICAL AEROFOIL≠)≠,
≠(≠IN A CIRCULATORY TRANSONIC POTENTIAL FLOW≠)≠,
≠(≠BY USING LIGHTHILLS 2ND INTEGRAL OPERATOR≠)≠).,


OUTPUT(41,≠(≠11S,D.4D3B,5S,+D.6D3B,6S,D.6D,//,7S,D.4D3B,11S,D.4D,//≠)≠,
≠(≠EPSILON(0)=≠)≠,EPSZERO,≠(≠ALFA=≠)≠,ALFA,≠(≠GAMMA=≠)≠,2*PI*GAMMA,
≠(≠TAU(1)=≠)≠,TI,≠(≠TAU(ZETA1)=≠)≠,TZ1).,
OUTPUT(41,≠(≠≠(≠CASE≠)≠,ZZDBB,≠(≠M-INF =≠)≠,ZD.4D,///≠)≠,
CASE,SQRT(5*TI/(1-TI))).,
OUTPUT(41,≠(≠≠(≠LAMBDA1 =≠)≠,+ZD.DD4B,≠(≠LAMBDA2 =≠)≠,+ZD.DD4B,///≠)≠,
L43,L45).,
R.=1.,I.=2.,LOWER.=1.,UPPER.=2.,FACT(/1/).=1.,FACT(/2/).=PI.,
                PD2.=FACT(/2/)*.5.,LNEPS0.=LN(EPSZERO).,
REWIND(43)., TAPE(TI,PSIT).,
≠FOR≠N.=0≠STEP≠ 1 ≠UNTIL≠ MAX ≠DO≠ ≠FOR≠ J.=1,2,3,4,5 ≠DO≠
PSITI(/N,J,0/).=PSIT(/N,J,0/)+2*TI*PSIT(/N,J,1/).,
REWIND(43)., RECO.=0., JP.=-1.,
≠FOR≠ N.=2,3 ≠DO≠
≠BEGIN≠ AB(/N/).=1/(N-1)*PSITI(/N,3,0/)*(FGAM(/1/)*LARRAY(/1,1,N,1/)+
  FGAM(/2/)*LARRAY(/2,1,N,1/)-FGAM(/3/)*(1-1/N)*EPSZERO≠POWER≠N*
  SIN(2*N*ALFA)).,
  AB(/N/).=AB(/N/)+L43*PI*FGAM(/3/)*C(/N/)*PSITI(/N,1,0/).,
  BB(/N/).=-1/(N-1)*PSITI(/N,3,0/)*(FGAM(/1/)*LARRAY(/1,1,N,2/)+
    FGAM(/2/)*LARRAY(/2,1,N,2/)-FGAM(/3/)*(1-1/N)*(1-EPSZERO≠POWER≠N*
    COS(2*N*ALFA))).,
  BB(/N/).=BB(/N/)+L45*FGAM(/3/)*C(/N/)*PSITI(/N,2,0/).,
≠END≠.,
≠IF≠ CUSP ≠THEN≠
≠BEGIN≠
COMPC.,  CHAPLYGIN(T,PSIT).,  FTCONV.,
J.=≠IF≠T≠GREATER≠TZ1+.00001≠THEN≠2≠ELSE≠1.,
RL(/1/).=-SOM(1,2,P(4)).,
RL(/2/).=-SOM(1,2,DPDT(/K/)).,
RL(/3/).=-SOM(1,2,DPDTH(/K/)).,
≠BEGIN≠ ≠REAL≠ DET,M33,M23,M13,A1,A2,A3,B1,B2,B4.,
≠ARRAY≠ P,Q(/1..3/),M(/1..3,1..4/).,
≠FOR≠RI.=1,2,3,4≠DO≠≠BEGIN≠M(/1,RI/).=ESOM(RI,PE( 5 )).,
M(/2,RI/).=ESOM(RI,PHITE(/K/)).. M(/3,RI/).=ESOM(RI,PHITHE(/K/)) ≠END≠.,
```

```
OUTPUT(41,≠(≠32S,//,7S,D.6D3B,9S,+D.6D,/≠)≠,
≠(≠CORRECTION FUNCTION QUANTITIES..≠)≠,≠(≠TAU(C)=≠)≠,T,
≠(≠THETA(C)=≠)≠,-TH*RAD).,
OUTPUT(41,≠(≠/,4(+.4D≠+D2B),+D.8D≠)≠,
M(/1,1/),M(/1,2/),M(/1,3/),M(/1,4/),RL(/1/)).,
OUTPUT(41,≠(≠/,4(+.4D≠+D2B),+D.8D≠)≠,
M(/2,1/),M(/2,2/),M(/2,3/),M(/2,4/),RL(/2/)).,
OUTPUT(41,≠(≠/,4(+.4D≠+D2B),+D.8D≠)≠,
M(/3,1/),M(/3,2/),M(/3,3/),M(/3,4/),RL(/3/)).,
M33.=M(/1,1/)*M(/2,2/)-M(/1,2/)*M(/2,1/).,
M23.=M(/1,1/)*M(/3,2/)-M(/1,2/)*M(/3,1/).,
M13.=M(/2,1/)*M(/3,2/)-M(/2,2/)*M(/3,1/).,
DET.=M(/1,3/)*M13-M(/2,3/)*M23+M(/3,3/)*M33.,
P(/3/).=(RL(/1/)*M13-RL(/2/)*M23+RL(/3/)*M33)/DET.,
RL(/1/).=RL(/1/)-M(/1,3/)*P(/3/)., RL(/2/).=RL(/2/)-M(/2,3/)*P(/3/).,
P(/2/).=(M(/1,1/)*RL(/2/)-M(/2,1/)*RL(/1/))/M33.,
P(/1/).=(RL(/1/)-M(/1,2/)*P(/2/))/M(/1,1/).,
Q(/3/).=(M(/1,4/)*M13-M(/2,4/)*M23+M(/3,4/)*M33)/DET.,
M(/1,4/).=M(/1,4/)-M(/1,3/)*Q(/3/).,
M(/2,4/).=M(/2,4/)-M(/2,3/)*Q(/3/).,
Q(/2/).=(M(/1,1/)*M(/2,4/)-M(/2,1/)*M(/1,4/))/M33.,
Q(/1/).=(M(/1,4/)-M(/1,2/)*Q(/2/))/M(/1,1/).,
B4.=1.0/EXPSI/EXPSI.,
A1.=CE(/3,2,2/)*B4.,
A2.=CE(/3,2,1/)*B4.,A3.=-B4., B1.=-CE(/3,2,1/)*B4., B2.=CE(/3,2,2/)*B4.,
≠IF≠ MU2IN ≠THEN≠
MULT(/4/).=-(BB(/2/)+B1*P(/1/)+B2*P(/2/)-MU2*(AB(/2/)+A1*P(/1/)+A2*
P(/2/)+A3*P(/3/))))/(MU2*(A1*Q(/1/)+A2*Q(/2/)+A3*Q(/3/))-(B1*Q(/1/)+B2*
Q(/2/)-B4))
≠ELSE≠
MULT(/4/).=(P(/1/)*Q(/1/)+P(/2/)*Q(/2/)+P(/3/)*Q(/3/))/
(Q(/1/)*Q(/1/)+Q(/2/)*Q(/2/)+Q(/3/)*Q(/3/)+1).,
MULT(/1/).=P(/1/)-MULT(/4/)*Q(/1/).,
MULT(/2/).=P(/2/)-MULT(/4/)*Q(/2/).,
MULT(/3/).=P(/3/)-MULT(/4/)*Q(/3/).,
OUTPUT(41,≠(≠/,4(+.4D≠+D2B)≠)≠,MULT(/1/),MULT(/2/),MULT(/3/),MULT(/4/))
≠END≠.,
≠FOR≠SIDE.=1,2≠DO≠ ≠FOR≠ SIG.=0,1 ≠DO≠
≠BEGIN≠ K.=(-1)≠POWER≠(SIDE+SIG).,
DX(/1,SIDE,SIG/).=DX(/1,SIDE,SIG/)+K*(MULT(/1/)*XTI(/1/)+MULT(/2/)*
  XTI(/2/)).,
DY(/1,SIDE,SIG/).=DY(/1,SIDE,SIG/)+K*(MULT(/1/)*YTI(/1/)+MULT(/2/)*
  YTI(/2/)).,
K.=(-1)≠POWER≠SIDE., ≠FOR≠J.=2,3≠DO≠ ≠BEGIN≠
DX(/J,SIDE,SIG/).=DX(/J,SIDE,SIG/)+K*(MULT(/1/)*XPI(/1/)
-MULT(/2/)*XPI(/2/)).,
DY(/J,SIDE,SIG/).=DY(/J,SIDE,SIG/)-K*(MULT(/1/)*YPI(/1/)
-MULT(/2/)*YPI(/2/)).,
≠END≠  ≠END≠.,
≠END≠ CUSP-TEST.,
OUTPUT(41,≠(≠//,23S≠)≠,≠(≠INTEGRATION CONSTANTS..≠)≠).,
≠FOR≠ SIG.=0,1 ≠DO≠ ≠BEGIN≠ OUTPUT(41,≠(≠/≠)≠).,
≠FOR≠ J.=1,2,3 ≠DO≠ ≠BEGIN≠ OUTPUT(41,≠(≠/,2S,D3B≠)≠,≠(≠J=≠)≠,J).,
≠FOR≠ SIDE .=1,2 ≠DO≠ ≠BEGIN≠ OUTPUT(41,≠(≠2(+ZD.5D2B)≠)≠,
DX(/J,SIDE,SIG/),DY(/J,SIDE,SIG/)) ≠END≠ ≠END≠ ≠END≠.,
≠BEGIN≠ ≠COMMENT≠ BEREKENING VAN DE KROMTESTRAAL.,
≠REAL≠ EPS,R0., ≠ARRAY≠ A,B,A1(/2..3/).,
≠FOR≠ N.=2,3 ≠DO≠
```

```
≠BEGIN≠ A(/N/).=
AB(/N/)+(≠IF≠ CUSP ≠THEN≠ EXP(-N*SI)*
(MULT(/1/)*CE(/3,N,2/)+MULT(/2/)*CE(/3,N,1/))≠ELSE≠ 0).,
B(/N/).=BB(/N/)+ (≠IF≠CUSP≠THEN≠EXP(-N*SI)*
(-MULT(/1/)*CE(/3,N,1/)+MULT(/2/)*CE(/3,N,2/))≠ELSE≠ 0) ≠END≠.,
A(/2/).=A(/2/)+(≠IF≠ CUSP ≠THEN≠  -MULT(/3/)*EXP(-2*SI) ≠ELSE≠ 0).,
B(/2/).=B(/2/)+(≠IF≠ CUSP ≠THEN≠ MULT(/4/)*EXP(-2*SI) ≠ELSE≠ 0).,
EPS.=-.5*ARCTAN(B(/2/)/A(/2/)).,  R0.=A(/2/)*A(/2/)+B(/2/)*B(/2/).,
≠FOR≠ N.=2,3 ≠DO≠ A1(/N/).=A(/N/)*COS(N*EPS)-B(/N/)*SIN(N*EPS).,
R0.=4*ABS(A1(/2/)/A1(/3/)*SQRT(TI*R0)).,
OUTPUT(41,≠(≠///,≠(≠STAGNATION POINT≠)≠,/≠)≠).,
OUTPUT(41,≠(≠B,≠(≠THETA≠)≠,8B,≠(≠X≠)≠,9B,≠(≠Y≠)≠,8B,≠(≠1/R≠)≠,8B,
≠(≠CP≠)≠≠)≠).,
OUTPUT(41,≠(≠//,3(+D.5DBB),.5D≠+ZDBB,+D.4D≠)≠,
-EPS+SIGN(EPS)*0.5*PI,-DX(/1,2,1/),DY(/1,2,1/),1/R0,
(1-TI)/TI/3.5*((1.0/(1-TI))≠POWER≠3.5-1)).,
OUTPUT(41,≠(≠//,≠(≠MU2=≠)≠≠)≠,B(/2/)/A(/2/)).,
≠END≠.,
≠BEGIN≠≠REAL≠ A,B,C,TS,TZ,PSI,TB,DT,TMAX,MAXDT,MINDT,THETAB,THETA,
        DTHETAB,DTHETA.,
  ≠INTEGER≠ W,MODE,FR,SDPSI,VT,TASK,NTH.,
  ≠BOOLEAN≠ TLPNT,SONIC.,
  ≠ARRAY≠ ERROR(/1..2/),TW,THW(/0..2/).,

≠PROCEDURE≠ PARAB(X,Y)., ≠VALUE≠ X,Y., ≠ARRAY≠ X,Y.,
≠BEGIN≠≠COMMENT≠ BEPAALT COEFFICIENTEN A,B EN C ALS Y(/I/)=AXX+BX+C,
  I=0,1,2.,
≠REAL≠ D.,
  D.=(Y(/2/)-Y(/0/))/(X(/2/)-X(/0/)).,
  A.=((Y(/1/)-Y(/0/))/(X(/1/)-X(/0/))-D)/(X(/1/)-X(/2/)).,
  B.=D-A*(X(/0/)+X(/2/)).,
  C.=Y(/0/)-X(/0/)*(A*X(/0/)+B)
≠END≠.,

≠PROCEDURE≠ ZERO(X, A, FA, B, FB, FX, E)., ≠VALUE≠ A,B.,
≠REAL≠ X, A, FA, B, FB, FX., ≠ARRAY≠ E.,
≠BEGIN≠ ≠REAL≠ C, FC, M, I, TOL, RE, AE.,
  ≠INTEGER≠ K.,
    RE.=E(/1/)., AE.=E(/2/)., K.=0.,
    X.=B., ≠GOTO≠ ENTRY.,
GOON.. K.=K+1., ≠IF≠ ABS(I-B) ≠LESS≠ TOL ≠THEN≠ I.=B+SIGN(C-B)*TOL.,
    X.=≠IF≠ SIGN(I-M) ≠EQUAL≠SIGN(B-I) ≠THEN≠ I ≠ELSE≠ M.,
    A.=B., FA.=FB., B.=X., FB.=FX.,
    ≠IF≠ SIGN(FC) ≠EQUAL≠ SIGN(FB) ≠THEN≠
ENTRY.. ≠BEGIN≠ C.=A., FC.=FA., ≠END≠.,
    ≠IF≠ ABS(FB) ≠GREATER≠ ABS(FC) ≠THEN≠
    ≠BEGIN≠ A.=B., FA.=FB., B.=C., FB.=FC., C.=A., FC.=FA., ≠END≠.,
    M.= (B+C) / 2.,
    I.=≠IF≠ FB-FA ≠NOTEQUAL≠ 0 ≠THEN≠ (A*FB-B*FA)/(FB-FA) ≠ELSE≠ M.,
    TOL.= ABS(B*RE)+AE.,
    ≠IF≠ ABS(M-B) ≠GREATER≠ TOL ≠AND≠ K ≠LESS≠ MAXIT≠THEN≠ ≠GOTO≠ GOON.,
    X.=B., TZ.=K.,
≠END≠ ZERO.,

≠PROCEDURE≠ ZEROSTAT(X,XB,ABSDX,FX,S,E,N).,
≠VALUE≠ XB,ABSDX,E,N.,
≠REAL≠ X,XB,ABSDX,FX., ≠INTEGER≠ S,N., ≠ARRAY≠ E.,
≠BEGIN≠≠REAL≠ XMK,FMK,DISCR,FM,XMU,MU,MIN.,
```

```
     ≠INTEGER≠ I,J,K,M., ≠BOOLEAN≠ ROOT.,
     ≠ARRAY≠ XI,FI,ABSX(/0..2/).,
     K.=2.,
     XI(/1/).=X.=XB., FI(/1/).=FX.,
     XI(/0/).=X.=XB-S*ABSDX*SIGN(FI(/1/))., FI(/0/).=FX.,
     XMU.=(XI(/0/)*FI(/1/)-XI(/1/)*FI(/0/))/(FI(/1/)-FI(/0/)).,
     ≠IF≠ S=SIGN((FI(/1/)-FI(/0/))/(XI(/1/)-XI(/0/))) ≠AND≠
       (XMU-XI(/0/))/(XI(/1/)-XI(/0/)) ≠GREATER≠ 0 ≠THEN≠ ≠GOTO≠ PROCZERO
       ≠ELSE≠ MU.=-1.0.,
     XMK.=-XB., FMK.=≠-20.,
   NEW PARAB..
     K.=K+1., ≠IF≠ K ≠GREATER≠N≠THEN≠≠BEGIN≠ S.=N., ≠GOTO≠ ENDPROC ≠END≠.,
     X.=XI(/2/).=(1.0-MU)*XI(/0/)+MU*XI(/1/)., FI(/2/).=FX.,
     PARAB(XI,FI).,
     DISCR.=B*B-4.0*A*C., ROOT.=0 ≠LESS≠ DISCR.,
     ≠IF≠ ROOT ≠THEN≠ XMU.=.5*(-B+S*SQRT(DISCR))/A ≠ELSE≠
     ≠BEGIN≠ XMU.=-.5*B/A., FM.=-.25*DISCR/A.,
       ≠IF≠ ABS(XMK/XMU-1) ≠LESS≠ F(/2/) ≠OR≠ ABS(FMK/FM-1) ≠LESS≠
         F(/2/) ≠THEN≠≠BEGIN≠ X.=XMU., S.=0., ≠GOTO≠ ENDPROC ≠END≠.,
       XMK.=XMU., FMK.=FM.,
     ≠END≠.,
     ≠COMMENT≠ REARRANGEMENT OF THE (XI,FI) TO DECREASING VALUES OF
       ABS(XI-XMU).,
     ≠FOR≠ I.=0,1,2 ≠DO≠ ABSX(/I/).=ABS(XI(/I/)-XMU).,
     ≠FOR≠ I.=0,1 ≠DO≠
     ≠BEGIN≠ MIN.=ABSX(/I/)., M.=I., ≠FOR≠ J.=I+1 ≠STEP≠ 1 ≠UNTIL≠ 2 ≠DO≠
       ≠IF≠ ABSX(/J/) ≠LESS≠ MIN ≠THEN≠
       ≠BEGIN≠ MIN.=ABSX(/J/)., M.=J ≠END≠.,
       MIN.=ABSX(/I/)., ABSX(/I/).=ABSX(/M/)., ABSX(/M/).=MIN.,
       MIN.=FI(/I/)., FI(/I/).=FI(/M/)., FI(/M/).=MIN.,
       MIN.=XI(/I/)., XI(/I/).=XI(/M/)., XI(/M/).=MIN
     ≠END≠.,
     MU.=(XMU-XI(/0/))/(XI(/1/)-XI(/0/)).,
     ≠IF≠ ≠NOT≠ ROOT ≠OR≠ MU ≠LESS≠ 0 ≠OR≠ SIGN(FI(/0/))=SIGN(FI(/1/))
     ≠THEN≠≠BEGIN≠ MU.=.25*(3*SIGN(MU)-1)., ≠GOTO≠ NEW PARAB ≠END≠.,
   PROCZERO..  ZERO(X,XI(/0/),FI(/0/),XI(/1/),FI(/1/),FX,E).,
    ENDPROC..  TS.=K.,
   ≠END≠ ZEROSTAT.,

   ≠REAL≠ ≠PROCEDURE≠ PSIBC.,
   ≠BEGIN≠≠REAL≠ PSI.,
   TH.=FACT(/3/).=-THETA.,
   SIG.=0.5*(1+SIGN(THETA)).,
   ≠IF≠ MODE=4 ≠THEN≠≠FOR≠ TEL.=1,2,3,4,5 ≠DO≠
   ≠BEGIN≠≠FOR≠ K.=1 ≠STEP≠ 1 ≠UNTIL≠ 48 ≠DO≠ LMAX(/TEL,K/).=0.,
   ≠IF≠ CUSP ≠THEN≠≠FOR≠ K.=3,4,5,8,9,10 ≠DO≠ ELMAX(/TEL,K/).=0
   ≠END≠.,
   PSI.=SOM(J,SIDE,P(MODE))+CORR(PE(MODE)).,
   OUTPUT(41,≠(≠/,+2ZD.4DB,+ZD.5DB≠)≠,THETA*RAD,PSI).,
   ≠IF≠ MODE=1 ≠THEN≠≠GOTO≠ ASSIGN.,
   X.=SOM(J,SIDE,XP(/K/))+CORR(XE(/K/))+DX(/J,SIDE,SIG/).,
   Y.=SOM(J,SIDE,YP(/K/))+CORR(YE(/K/))+DY(/J,SIDE,SIG/).,
     ≠IF≠ SONIC ≠THEN≠
     ≠BEGIN≠ OUTPUT(41,≠(≠2(+ZD.5DB)≠)≠,-X,Y).,
             ≠GOTO≠ ASSIGN
     ≠END≠.,
   PHIT.=SOM(J,SIDE,DPDT(/K/))+CORR(PHITE(/K/)).,
   PHITH.=SOM(J,SIDE,DPDTH(/K/))+CORR(PHITHE(/K/)).,
```

```
≠IF≠ TLPNT ≠THEN≠ JP.=0 ≠ELSE≠
JP.=-((1-T)≠POWER≠2.5*T/TI)≠POWER≠2/((2*T*PHIT)≠POWER≠2+
(1-6*T)/(1-T)*PHITH≠POWER≠2).,
DXDTH.=FT*(PHIT*COS(TH)-.5/T*PHITH*SIN(TH)).,
DYDTH.=FT*(PHIT*SIN(TH)+.5/T*PHITH*COS(TH)).,
RR.=-JP/(1+(1-MACH*MACH)*(PHITH/(2*T*PHIT))≠POWER≠2)*TI/T.,
OUTPUT(41,≠(≠4(+ZD.5DB),B,+.5D≠+ZD,//,
+ZD.5D11B,2(+ZD.5DB),21B,+.5D≠+ZD,//≠)≠,
-X,Y,PHIT,-PHITH,JP,-TH,DXDTH,-DYDTH,SQRT(ABS(RR))).,
≠BEGIN≠ ≠INTEGER≠ MAX,KMAX., OUTPUT(41,≠(≠≠(≠  K   MAX   ≠)≠≠)≠).,
≠FOR≠ TEL.=1,2,3,4,5 ≠DO≠
≠BEGIN≠ MAX.=0., ≠FOR≠ K.=1 ≠STEP≠ 1 ≠UNTIL≠ 48 ≠DO≠
≠BEGIN≠ ≠IF≠ LMAX(/TEL,K/) ≠GREATER≠ MAX ≠THEN≠
≠BEGIN≠ KMAX.=K.,  MAX.=LMAX(/TEL,K/) ≠END≠ ≠END≠.,
OUTPUT(41,≠(≠2BZD,B2ZD2B≠)≠,KMAX,MAX) ≠END≠.,
≠IF≠ CUSP ≠THEN≠ ≠BEGIN≠
OUTPUT(41,≠(≠//,≠(≠  KC MAX   ≠)≠≠)≠).,
≠FOR≠ TEL.=1,2,3,4,5 ≠DO≠
≠BEGIN≠ MAX.=0., ≠FOR≠ K.=3,4,5,8,9,10 ≠DO≠
≠BEGIN≠ ≠IF≠ ELMAX(/TEL,K/) ≠GREATER≠ MAX ≠THEN≠ ≠BEGIN≠ KMAX.=K.,
MAX.=ELMAX(/TEL,K/) ≠END≠ ≠END≠.,
OUTPUT(41,≠(≠2BZD,B2ZD2B≠)≠,KMAX,MAX) ≠END≠ ≠END≠ ≠END≠.,
≠IF≠ 0 ≠LESS≠ JP ≠THEN≠ ≠BEGIN≠ OUTPUT(41,≠(≠//,≠(≠LIMIT-LINE≠)≠≠)≠).,
≠GOTO≠ READ TASK ≠END≠.,
ASSIGN.. PSIBC.=PSI
≠END≠ PSIBC..

≠PROCEDURE≠ NEWTAU.,
≠BEGIN≠≠INTEGER≠ KT.,
KT.=100*T.,
≠IF≠ T ≠LESS≠ 0.05 ≠AND≠ ABS(100*T-KT) ≠GREATER≠ ≠-8 ≠THEN≠
CHAPLYGIN(T,PSIT) ≠ELSE≠ TAPE(T,PSIT).,
FTCONV., MACH.=SQRT(5*T/(1-T)).,
CP.=(1-TI)/TI/3.5*(((1-T)/(1-TI))≠POWER≠3.5-1).,
OUTPUT(41,≠(≠*≠)≠).,
OUTPUT(41,≠(≠//,4S,D.6D4B,2S,D.4D4B,3S,+D.4D,//,4S,.6D5B,2S,D,/≠)≠,
≠(≠TAU=≠)≠,T,≠(≠M=≠)≠,MACH,≠(≠CP=≠)≠,CP,≠(≠TOL=≠)≠,TOL,≠(≠J=≠)≠,J).,
SIG.=0.5*(1+SIGN(THETAB)).,
≠IF≠ TLPNT ≠THEN≠ OUTPUT(41,≠(≠///25B,≠(≠TAIL POINT≠)≠≠)≠) ≠ELSE≠
≠BEGIN≠
≠IF≠ SIDE=LOWER ≠THEN≠
OUTPUT(41,≠(≠///25B6S≠)≠,≠(≠LOWER ≠)≠) ≠ELSE≠
OUTPUT(41,≠(≠///25B6S≠)≠,≠(≠UPPER ≠)≠).,
≠IF≠ SONIC ≠THEN≠ OUTPUT(41,≠(≠≠(≠SONIC LINE≠)≠≠)≠) ≠ELSE≠
≠BEGIN≠ ≠IF≠ FR=2 ≠THEN≠ OUTPUT(41,≠(≠≠(≠REAR PART≠)≠≠)≠) ≠ELSE≠
  OUTPUT(41,≠(≠≠(≠FRONT PART≠)≠≠)≠).,
≠END≠.,
≠END≠.,
≠IF≠ SONIC ≠THEN≠
OUTPUT(41,≠(≠//,3B,5S,6B,3S,8B,S,9B,S,/≠)≠,
≠(≠THETA≠)≠,≠(≠PSI≠)≠,≠(≠X≠)≠,≠(≠Y≠)≠) ≠ELSE≠
≠BEGIN≠
OUTPUT(41,≠(≠//,3B,5S,6B,3S,8B,S,9B,S,4B,9S,B,11S,3B,5S,/≠)≠,
≠(≠THETA≠)≠,≠(≠PSI≠)≠,≠(≠X≠)≠,≠(≠Y≠)≠,≠(≠DPSI/DTAU≠)≠,≠(≠DPSI/DTHETA≠)≠,
≠(≠DET J≠)≠).,
OUTPUT(41,≠(≠20B,19S,25B,3S,/≠)≠,≠(≠DX/DTHETA DY/DTHETA≠)≠,≠(≠1/R≠)≠).,
≠END≠
≠END≠ NEWTAU.,
```

```

READ TASK..
EOF(40,EOP).,
INPUT(40,≠(≠/≠)≠).,
INCHARACTER(40,≠(≠LUT≠)≠,TASK).,
≠IF≠ TASK=0 ≠THEN≠
≠BEGIN≠ OUTPUT(41,≠(≠//,≠(≠ILLEGAL TASK≠)≠≠)≠)., ≠GOTO≠ EOP ≠END≠.,
REWIND(43).,
≠IF≠ TASK=3 ≠THEN≠ ≠GOTO≠ TAILPOINT.,
SIDE.=TASK.,
INCHARACTER(40,≠(≠FRS≠)≠,FR).,
≠IF≠ FR=0 ≠THEN≠
≠BEGIN≠ OUTPUT(41,≠(≠//,≠(≠ILLEGAL TASK≠)≠≠)≠)., ≠GOTO≠ EOP ≠END≠.,
≠IF≠ FR=3 ≠THEN≠ ≠GOTO≠ SONIC LINE.,

≠COMMENT≠ THE COMPUTATION OF AN AEROFOIL PART.,
TLPNT.=SONIC.= ≠FALSE≠.,
SDPSI.=-3+2*FR., MODE.=1.,
INPUT(40,≠(≠≠)≠,TB,DT,TMAX,MAXDT,MINDT,THETAB,DTHETAB,DTHETA).,
THETAB.=THETAB/RAD., DTHETAB.=DTHETAB/RAD., DTHETA.=DTHETA/RAD.,
ERROR(/1/).=ERROR(/2/).=TOL.,
≠FOR≠ W.=0,1,2 ≠DO≠
≠BEGIN≠ TW(/W/).=T.=TB+W*DT.,
≠IF≠ T ≠GREATER≠ TMAX+≠-4 ≠THEN≠ ≠GOTO≠ READ TASK.,
NEWTAU.,
ZEROSTAT(THETA,THETAB,DTHETAB,PSIBC,SDPSI,ERROR,MAXIT).,
≠IF≠ SDPSI=0 ≠OR≠ SDPSI =MAXIT ≠THEN≠ ≠GOTO≠ READ TASK.,
THW(/W/).=THETAB.=THETA.,
MODE.=4., PSI.=PSIBC., MODE.=1.,
≠END≠.,
≠GOTO≠ PARABOLA.,
NEXT.. T.=TW(/2/).=T+DT.,
≠IF≠ T ≠GREATER≠ TMAX+≠-4 ≠THEN≠ ≠GOTO≠ READ TASK.,
NEWTAU., MODE.=1.,
ZEROSTAT(THETA,THETAB,DTHETA,PSIBC,SDPSI,ERROR,MAXIT).,
THW(/2/).=THETA.,
≠IF≠ ABS(SDPSI)=1 ≠THEN≠
≠BEGIN≠ MODE.=4., PSI.=PSIBC ≠END≠
≠ELSE≠ ≠GOTO≠ READ TASK.,
PARABOLA..
PARAB(THW,TW).,
    ≠IF≠ -A ≠LESS≠ ≠-10 ≠THEN≠ DT.=1.0 ≠ELSE≠
DT.=ABS(0.25*(2.0*A*THW(/2/)+B) ≠POWER≠ 2/(4.0*A)).,
DT.=≠IF≠ MAXDT ≠LESS≠ DT ≠THEN≠ MAXDT ≠ELSE≠≠IF≠ DT ≠LESS≠ MINDT
≠THEN≠ MINDT ≠ELSE≠ DT.,
VT.=DT/MINDT.. DT.=VT*MINDT.,
THW(/0/).=THW(/1/)., THW(/1/).=THW(/2/).,
TW(/0/).=TW(/1/).. TW(/1/).=TW(/2/).,
THETAB.=THW(/1/)+DT*(THW(/1/)-THW(/0/))/(TW(/1/)-TW(/0/)).,
≠GOTO≠ NEXT.,

TAILPOINT..
TLPNT.=≠TRUE≠.,  SONIC.=≠FALSE≠.,
INPUT(40,≠(≠B≠)≠,T,THETAB).,
THETA.=THETAB/RAD.,  TH.=-THETA.,
MODE.=4.,  SIDE.=(3-SIGN(THETA))/2.,
NEWTAU.,
PSI.=PSIBC.,
```

```
       ≠GOTO≠ READ TASK.,

       SONIC LINE..
       T.=1/6.,  SONIC.=≠TRUE≠., MODE.=3.,
       TLPNT.=≠FALSE≠.,
       INPUT(40,≠(≠≠)≠,NTH,THETAB).,  THETA.=THETAB.,
760**  NEWTAU.,
       ≠FOR≠ TTH.=1 ≠STEP≠ 1 ≠UNTIL≠ NTH ≠DO≠
       ≠BEGIN≠
         ≠IF≠ TTH ≠GREATER≠ 1 ≠THEN≠ INPUT(40,≠(≠≠)≠,THETA).,
         THETA.=THETA/RAD.,
         PSI.=PSIBC.,
       ≠END≠ TTH-CYCLE.,
       ≠GOTO≠ READ TASK.,
       ≠END≠.,
       EOP..
770**  ≠END≠.,
       ≠END≠
                 ≠EOP≠
```

```
          LINE    0        PROGRAM BEGINS       (MESSAGE)          1
          LINE  771        PROGRAM ENDS         (MESSAGE)          1
          LINE  771        SOURCE DECK ENDS     (MESSAGE)          1
          LINE  425        NON-FORMAT STRING    (MESSAGE)          1
          LINE  426        NON-FORMAT STRING    (MESSAGE)          1
          LINE  427        NON-FORMAT STRING    (MESSAGE)          1
          LINE  431        NON-FORMAT STRING    (MESSAGE)          1
          LINE  431        NON-FORMAT STRING    (MESSAGE)          1
          LINE  431        NON-FORMAT STRING    (MESSAGE)          1
          LINE  432        NON-FORMAT STRING    (MESSAGE)          1
          LINE  432        NON-FORMAT STRING    (MESSAGE)          1
          LINE  465        NON-FORMAT STRING    (MESSAGE)          1
          LINE  465        NON-FORMAT STRING    (MESSAGE)          1
          LINE  466        NON-FORMAT STRING    (MESSAGE)          1
          LINE  514        NON-FORMAT STRING    (MESSAGE)          1
          LINE  516        NON-FORMAT STRING    (MESSAGE)          1
          LINE  674        NON-FORMAT STRING    (MESSAGE)          1
          LINE  674        NON-FORMAT STRING    (MESSAGE)          1
          LINE  674        NON-FORMAT STRING    (MESSAGE)          1
          LINE  674        NON-FORMAT STRING    (MESSAGE)          1
          LINE  674        NON-FORMAT STRING    (MESSAGE)          1
          LINE  679        NON-FORMAT STRING    (MESSAGE)          1
          LINE  680        NON-FORMAT STRING    (MESSAGE)          1
          LINE  688        NON-FORMAT STRING    (MESSAGE)          1
          LINE  688        NON-FORMAT STRING    (MESSAGE)          1
          LINE  688        NON-FORMAT STRING    (MESSAGE)          1
          LINE  688        NON-FORMAT STRING    (MESSAGE)          1
          LINE  691        NON-FORMAT STRING    (MESSAGE)          1
          LINE  691        NON-FORMAT STRING    (MESSAGE)          1
          LINE  691        NON-FORMAT STRING    (MESSAGE)          1
          LINE  691        NON-FORMAT STRING    (MESSAGE)          1
          LINE  691        NON-FORMAT STRING    (MESSAGE)          1
          LINE  691        NON-FORMAT STRING    (MESSAGE)          1
          LINE  692        NON-FORMAT STRING    (MESSAGE)          1
          LINE  693        NON-FORMAT STRING    (MESSAGE)          1
          LINE  700        NON-FORMAT STRING    (MESSAGE)          1
          LINE  706        NON-FORMAT STRING    (MESSAGE)          1

               THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE     F.I.L.X


CORE MAP  12.43.31.  NORMAL          CONTROL                                    000100    063254    061054    002200
      ---TIME---LOAD MODE --L1--L2-----TYPE---------------USER---+---CALL-------------FWA LOAD--LWA LOAD--BLNK COMN--LENGTH--
      FWA LOADER  103741  FWA TABLES  076625
      -PROGRAM----ADDRESS-                          --LABELED---COMMON--
      XXALGOL     000340                                   DATA       000100
      ALGORUN     042340                                   DATA       000100

      ALGLB00     044712                                   DATA       000100
      ALGLB01     050127                                   DATA       000100
      ALGLB02     050656                                   DATA       000100
      ALGLB03     056525                                   DATA       000100
      ALGLB05     057360                                   DATA       000100
      ALGLB06     060152                                   DATA       000100
      --ENTRY-----ADDRESS-                                        REFERENCES
      XXALGOL     041367
      ALGORUN     042340         XXALGOL

      ALGLB00     044712         XXALGOL

      ALGLB01     050127         XXALGOL

      ALGLB02     050656         XXALGOL

      ALGLB03     056530         XXALGOL

      ALGLB05     057360         XXALGOL

      ALGLB06     060152         XXALGOL

      ----UNSATISFIED EXTERNALS-----                              REFERENCES


CHANNEL.60=INPUT.P80.R
CHANNEL.61=OUTPUT.P136.PP60.R
CHANNEL.40=60
CHANNEL.41=61
CHANNEL.43=LU43.A.R
CHANNEL.END
```

```
≠BEGIN≠ ≠COMMENT≠ BLOK 1. PROGRAM T 320. SMOOTHING AND INTERPOLATION
OF THE FUNCTIONS. WHICH ARE SPECIFIED BY THE VALUES OF THE FUNCTION AND
ITS FIRST TWO DERIVATES AT A DISCRETE SET OF ORDINATES, USING..
A. SPLINE INTERPOLATION TECHNIQUES.
B. A LEAST SQUARES CONDITION AT THE GIVEN VALUES OF THE FUNCTION AND ITS
   FIRST TWO DERIVATES, AND
C. A SMOOTHNESS CONDITION ON THE THIRD DERIVATIVE..
≠INTEGER≠ NC, IC..
≠REAL≠ C1.C2.C3,Y0,Y1,Y2.Y3,Y4,Y5,A3.A4,A5..

START OF PROGRAM.. INREAL(40.IC)..
≠IF≠ IC ≠LESS≠ 0 ≠THEN≠ ≠GOTO≠ RUNS COMPLETED.. INREAL(40.NC)..

≠BEGIN≠ ≠COMMENT≠ BLOK 2..
≠INTEGER≠ I.J..
≠ARRAY≠ XC(/0..NC/), Y.YCORR(/0..3*NC+2/)..

≠PROCEDURE≠ SMOOTH THE VECTOR Y..
≠BEGIN≠ ≠REAL≠ EPS, NU0. NU1, NU2, MU3, MU4, MU5, SIGMA0, EPS1, EPS2,
  TOL1. TOL2. SIGMA1. SIGMA2. RHO3, RHO4. RHO5, E. S.
  C1. C2. C3. C4, C5. C6. C7. C8, C9. C10. C11. C12. C13. C14, C15..
  ≠INTEGER≠ I.J,K.L.M.N.O,P,Q. NSIGMA. NRHO. KC, KMAX, KKC. KKMAX..
  ≠ARRAY≠ EPSE. DC. SC, FC. YW(/0..3*NC+2/).
  R1(/0..3*NC+2. 4..9/). R2(/0..3*NC+2. 1..5/).
  LC(/1..6. 1..6/). D0.D1(/1..5/). R(/0..3*NC+2. 1..9/)..

  ≠BEGIN≠ ≠COMMENT≠ 1. READ AND PRINT INSTRUCTIONS FOR THE
    WEIGHT VECTORS E AND S..
    ≠PROCEDURE≠UTPUT(NSR.A.B,C.K.A1.B1.C1.AC.M)..≠VALUE≠NSR.A1.B1,C1..
    ≠REAL≠ A,B.C,A1.B1,C1.. ≠INTEGER≠ NSR,K.M.. ≠ARRAY≠ AC..
    ≠BEGIN≠ ≠FOR≠ I.=1 ≠STEP≠ 1 ≠UNTIL≠ NSR ≠DO≠
      ≠BEGIN≠ INPUT(40.≠(≠≠)≠.A.B.C,K)..
        OUTPUT(41,≠(≠/,3(+D.7D≠+ZZDB).+ZDB≠)≠. A.B.C,K)..
        ≠IF≠ NSR ≠NOTGREATER≠ 1 ≠THEN≠ K.=NC..
        ≠FOR≠ J.=J+1 ≠WHILE≠ J ≠NOTGREATER≠ K ≠DO≠
        ≠BEGIN≠ L.=3*J.. ≠IF≠ M=0 ≠THEN≠
          ≠BEGIN≠ AC(/L/).= A1*A.. AC(/L+1/).= B1*B.. AC(/L+2/).= C1*C..
          ≠END≠ ≠ELSE≠
          ≠BEGIN≠ AC(/L/).= A1/A.. AC(/L+1/).= B1/B.. AC(/L+2/).= C1/C..
          ≠END≠..
        ≠END≠.. J.= K..
      ≠END≠..
    ≠END≠ OF PROCEDURE UTPUT..

    INPUT(40,≠(≠≠)≠.
    I.EPS,NU0.NU1.NU2.MU3,MU4,MU5.NSIGMA,KKMAX.KMAX,TOL1,TOL2)..
    OUTPUT(41,≠(≠*,≠(≠DATA INPUT TAPE NUMBER B≠)≠.+ZD≠)≠,1)..
    OUTPUT(41,≠(≠7(/,3S.3B,≠(≠=≠)≠,+ZD.7D≠+ZZD)≠)≠.
    ≠(≠EPS≠)≠.EPS,≠(≠NU0≠)≠.NU0.≠(≠NU1≠)≠,NU1,≠(≠NU2≠)≠,NU2,
    ≠(≠MU3≠)≠.MU3.≠(≠MU4≠)≠.MU4,≠(≠MU5≠)≠,MU5)..
    OUTPUT(41.≠(≠3(/,6S,≠(≠=≠)≠,+ZD)≠)≠.
    ≠(≠NSIGMA≠)≠,NSIGMA,≠(≠KKMAX ≠)≠.KKMAX,≠(≠KMAX  ≠)≠,KMAX)..
    OUTPUT(41,≠(≠2(/,4S.3B,≠(≠=≠)≠,+D.7D≠+ZZD)≠)≠,
    ≠(≠TOL1≠)≠,TOL1,≠(≠TOL2≠)≠,TOL2)..
    OUTPUT(41,≠(≠/,≠(≠SIGMA0.SIGMA1.SIGMA2.K≠)≠≠)≠)..
    J.=-1..
    UTPUT(NSIGMA,SIGMA0.SIGMA1.SIGMA2,K,NU0.NU1,NU2.EC,1)..
    J.= -1.. NRHO.= 1..
```

APPENDIX E
LISTING OF SMOOTH

```
     OUTPUT(41,≠(≠/,≠(≠RHO3,RHO4,RHO5,K≠)≠≠)≠).,
      UTPUT(NRHO,RHO3,RHO4,RHO5,K,MU3,MU4,MU5,SC,0).,
   ≠END≠.,

   ≠COMMENT≠ 2. DETERMINATION OF THE MATRIX H.,
   ≠COMMENT≠ 2.1. DEFINITION OF THE MATRIX LC.,
   LC(/1,1/).=LC(/4,4/).= +720.,
   LC(/5,2/).=LC(/2,5/).= +168.,
   LC(/6,2/).=LC(/2,6/).= - 24.,
   LC(/4,1/).=LC(/1,4/).= -720.,
   LC(/3,3/).=LC(/6,6/).= +  9.,
   LC(/5,3/).=LC(/3,5/).= + 24.,
   LC(/2,2/).=LC(/5,5/).= +192.,
   LC(/6,3/).=LC(/3,6/).= -  3.,
   LC(/3,2/).=LC(/2,3/).= + 36.,
   LC(/6,5/).=LC(/5,6/).= - 36.,
   LC(/4,2/).=LC(/2,4/).=LC(/5,4/).=LC(/4,5/).= -360.,
   LC(/2,1/).=LC(/5,1/).=LC(/1,2/).=LC(/1,5/).= +360.,
   LC(/3,1/).=LC(/1,3/).=LC(/6,4/).=LC(/4,6/).= + 60.,
   LC(/6,1/).=LC(/4,3/).=LC(/3,4/).=LC(/1,6/).= - 60.,

   ≠COMMENT≠2.2.  CYCLE DETERMINING H.,
   KKC.= 0.,
   ONCE MORE 1.. KKC.= KKC+1.,
   OUTPUT(41,≠(≠*,≠(≠KKC=≠)≠,ZZD≠)≠,KKC).,
   ≠FOR≠ J.=0 ≠STEP≠ 1 ≠UNTIL≠ NC ≠DO≠
   ≠BEGIN≠ I.= 3*J-1., ≠IF≠ J ≠LESS≠ NC ≠THEN≠
     ≠BEGIN≠ C1.= C2.= 1/( XC(/J+1/)-XC(/J/)).,
       ≠FOR≠ K.=1 ≠STEP≠ 1 ≠UNTIL≠ 5 ≠DO≠
       ≠BEGIN≠ D0(/K/).= C2., C2.= C2*C1 ≠END≠.,
       ≠IF≠ KMAX=1 ≠AND≠ KKMAX=1 ≠THEN≠ ≠GOTO≠ KMAX ONE.,

       ≠COMMENT≠ 2.2.1. COMPUTATION OF RHOJ.,
       C1.= 0., ≠FOR≠ K.=1 ≠STEP≠ 1 ≠UNTIL≠ 6 ≠DO≠
       ≠BEGIN≠ C2.= C3.= 0., M.= 7-K.,
         ≠FOR≠ L.=K ≠STEP≠ 1 ≠UNTIL≠ 6 ≠DO≠
         C2.=C2+LC(/K,L/)*D0(/M-L+(≠IF≠ L ≠LESS≠ 4 ≠THEN≠ 0 ≠ELSE≠ 3)+
         (≠IF≠ K ≠LESS≠ 4 ≠THEN≠ 0 ≠ELSE≠ 3)/)*
         (≠IF≠ KKC=1 ≠THEN≠ Y(/I+L/) ≠ELSE≠ YCORR(/I+L/))*
         (≠IF≠ L ≠NOTEQUAL≠ K ≠THEN≠ 2.0 ≠ELSE≠ 1.0).,
         C1.=C1+C2*
         (≠IF≠ EC(/I+K/) ≠LESS≠ ≠-20 ≠AND≠ KKC=1 ≠THEN≠ 0 ≠ELSE≠
         (≠IF≠ KKC=1 ≠THEN≠ Y(/I+K/) ≠ELSE≠ YCORR(/I+K/))).,
         ≠IF≠ ABS(C1) ≠GREATER≠ C3 ≠THEN≠ C3.=ABS(C1)
       ≠END≠ OF K CYCLE.,
       ≠IF≠ C1 ≠LESS≠ C3*≠- 9 ≠THEN≠
       ≠BEGIN≠OUTPUT(41,≠(≠/,≠(≠FAILURE IN COMPUTATION OF RHO ≠)≠,
         +ZDB,(+D.7D≠+ZZDB)≠)≠, J,C1,C3).,
         C1.= ABS(C1).,
       ≠END≠.,
       ≠IF≠ C1 ≠LESS≠ ≠-100 ≠THEN≠ C1.=≠-100.,
       ≠FOR≠ K.=1,2,3 ≠DO≠ SC(/I+K/).=
       ≠IF≠ KKC=1 ≠THEN≠ SC(/I+K/)/C1 ≠ELSE≠ 1/C1.,
     ≠END≠ OF DETERMINATION OF RHOJ.,
     ≠COMMENT≠ 2.2.2. COMPUTATION OF FJ AND ITS CONTRIBUTION TO H.,
   KMAX ONE..
     ≠FOR≠ K.=1,2,3 ≠DO≠
     ≠BEGIN≠ M.=7-K., ≠IF≠ J ≠GREATER≠ 0 ≠THEN≠
```

```
        ≠BEGIN≠ C1.=SC(/I-3+K/).,
          ≠FOR≠ L.=K+3 ≠STEP≠ 1 ≠UNTIL≠ 6 ≠DO≠
          R1(/I+K,L/).=LC(/L,K+3/)*D1(/M-L+3/)*C1
        ≠END≠.,
        ≠IF≠ J ≠LESS≠ NC ≠THEN≠
        ≠BEGIN≠ C1.=SC(/I+K/).,
          ≠FOR≠ L.=K≠STEP≠ 1 ≠UNTIL≠ 6 ≠DO≠
          R1(/I+K,L+3/).=LC(/L,K/)*D0(/M-L+(≠IF≠L≠LESS≠4≠THEN≠0≠ELSE≠3)/)
          *C1+(≠IF≠L≠LESS≠4≠AND≠J≠GREATER≠0≠THEN≠R1(/I+K,L+3/)≠ELSE≠0)
        ≠END≠
      ≠END≠ OF DETERMINATION OF FJ AND ITS CONTRIBUTION TO H.,
      ≠FOR≠ K.=1 ≠STEP≠ 1 ≠UNTIL≠ 5 ≠DO≠ D1(/K/).=D0(/K/)
    ≠END≠ OF J CYCLE.,

    ≠COMMENT≠ 3. COMPUTATION OF E,S,EPS, AND THE MATRIX EPS*EC+H.,
    KC.=1., KMAX.=ABS(KMAX).,
    OUTPUT(41,≠(≠/,≠(≠K=  ≠)≠,+ZDB≠)≠,0).,
  AGAIN..
    ≠IF≠ KC ≠GREATER≠ 1 ≠THEN≠
    ≠BEGIN≠ OUTPUT(41,≠(≠/,≠(≠NUMBER OF ITERATIONS IN RESIDUAL ≠)≠,
      ≠(≠VECTOR METHOD..≠)≠,+ZDB,/≠)≠,I).,
      OUTPUT(41,≠(≠≠(≠TOLERANCE TESTS ARE ≠)≠≠)≠).,
      ≠IF≠ C8 ≠LESS≠ 0 ≠THEN≠ OUTPUT(41,≠(≠≠(≠NOT ≠)≠≠)≠).,
      OUTPUT(41,≠(≠≠(≠SATISFIED≠)≠≠)≠).,
    ≠END≠..

    S.=E.=C6.=C8.=0., IC.=O.=0.,
    ≠FOR≠ J.=0≠STEP≠ 1≠UNTIL ≠ NC ≠DO≠
    ≠BEGIN≠ N.=3*J.,
      Q.= ≠IF≠ J≠EQUAL≠0 ≠THEN≠ 4 ≠ELSE≠ 1.,
      I.=≠IF≠ J≠EQUAL≠ NC ≠THEN≠ 6 ≠ELSE≠ 9.,
      ≠FOR≠ K.=0,1,2 ≠DO≠
      ≠BEGIN≠ M.=N+K.,
        C3.= ≠IF≠ KC=1 ≠AND≠ KKC=1 ≠THEN≠ Y(/M/) ≠ELSE≠ YCORR(/M/).,
        C2.=C9.=0., C5.=EC(/M/).,
        ≠FOR≠ L.=I ≠STEP≠ -1 ≠UNTIL≠ Q ≠DO≠
        ≠BEGIN≠ P.=N+L-4.,
          C4.=≠IF≠KC=1 ≠AND≠ KKC=1 ≠THEN≠Y(/P/)≠ELSE≠ YCORR(/P/).,
          C1.=≠IF≠L≠GREATER≠3+K≠THEN≠R1(/M,L/)≠ELSE≠
          (≠IF≠L≠GREATER≠3≠THEN≠R1(/P,K+4/)≠ELSE≠R1(/P,K+7/)).,
          C7.=C1*C4., S.=S+C3*C7.,
          ≠IF≠ C8≠LESS≠ABS(S)≠THEN≠ C8.=ABS(S).,
          ≠IF≠ C5≠GREATER≠≠-10 ≠THEN≠
          ≠BEGIN≠ C2.=C2+C7.. ≠IF≠C9≠LESS≠ABS(C2)≠THEN≠C9.=ABS(C2) ≠END≠..
        ≠END≠.,
        ≠IF≠ C9 ≠GREATER≠ ≠+9*ABS(C2) ≠THEN≠
        ≠BEGIN≠C2.=0.. O.=O+1 ≠END≠.,
        ≠IF≠ C5≠GREATER≠≠-10 ≠THEN≠
        ≠BEGIN≠ ≠IF≠ C5 ≠LESS≠ ≠+20 ≠THEN≠ IC.=IC+1.. C6.=C6+C2*C2/C5.,
          C2.=Y(/M/)-C3.,≠IF≠C5≠LESS≠≠+16≠OR≠ABS(C2)≠GREATER≠≠-9≠THEN≠
          E.= E+ C2 * C2 * C5.,
        ≠END≠.,
      ≠END≠..
    ≠END≠..

    OUTPUT(41,≠(≠/,≠(≠E    =≠)≠,+D.7D≠+ZZD≠)≠,E).,
    OUTPUT(41,≠(≠/,≠(≠S    =≠)≠≠)≠).,
    ≠IF≠ ABS(C8/S)≠GREATER≠ ≠+9 ≠THEN≠ OUTPUT(41,≠(≠≠(≠NO SIGNIFICANT≠)≠,
```

```
      ≠(≠ ANSWER≠)≠≠)≠) ≠ELSE≠ OUTPUT(41•≠(≠+D.7D≠+ZZDB≠)≠•S)..             SMOOTH
      ≠IF≠ KC = 1 ≠THEN≠                                                   SMOOTH
      ≠BEGIN≠ OUTPUT(41•≠(≠/•≠(≠WEIGHT TABLE FOR THE RHO I≠)≠•/•≠(≠I≠)≠•    SMOOTH
        7B•≠(≠RHO I≠)≠≠)≠)..                                              SMOOTH
        ≠FOR≠J.=0≠STEP≠1≠UNTIL≠NC-1≠DO≠                                    SMOOTH
        OUTPUT(41•≠(≠/•+ZDB•+D.7D≠+ZZD≠)≠• J• SC(/3*J/))..                 SMOOTH
      ≠END≠..                                                              SMOOTH
      ≠IF≠(E/IC≠GREATER≠0.8≠AND≠E/IC≠LESS≠1.2)≠OR≠KC≠GREATER≠KMAX≠THEN≠     SMOOTH
      ≠BEGIN≠ OUTPUT(41•≠(≠/•≠(≠SMOOTHING COMPLETED AT K =≠)≠•+7D≠)≠•KC-1).. SMOOTH
        ≠GOTO≠ READY..                                                     SMOOTH
      ≠END≠..                                                              SMOOTH
                                                                           SMOOTH
      ≠IF≠KMAX=1≠THEN≠≠GOTO≠KMAXONEB..                                     SMOOTH
      EPS.=≠IF≠KC≠LESS≠3≠THEN≠SQRT(C6/IC)/(≠IF≠E≠LESS≠IC≠THEN≠100.0        SMOOTH
        ≠ELSE≠ 1.0)                                                        SMOOTH
      ≠ELSE≠ EPS*E/IC..                                                    SMOOTH
                                                                           SMOOTH
    KMAX ONE A..                                                           SMOOTH
      OUTPUT(41•≠(≠*•≠(≠K   =≠)≠•+ZZD≠)≠• KC)..                            SMOOTH
      ≠IF≠KMAX=1≠THEN≠≠GOTO≠KMAX ONE B..                                   SMOOTH
      OUTPUT(41•≠(≠/•≠(≠EPS    =≠)≠•+.12D≠+ZZDB•/•≠(≠I    =≠)≠•+ZZD•/•      SMOOTH
      ≠(≠C6   =≠)≠•+.12D≠+ZZD•/•≠(≠NUMBER OF NON SIGNIFICANT ≠)≠•           SMOOTH
      ≠(≠CONTRIBUTIONS IN C6 AND E..≠)≠•+ZZD≠)≠•                           SMOOTH
      EPS• IC• C6• Q)..                                                    SMOOTH
                                                                           SMOOTH
    KMAX ONE B.. Q.= 3*NC+2.. ≠IF≠ KMAX = 1 ≠THEN≠                         SMOOTH
      ≠BEGIN≠≠FOR≠L.=0≠STEP≠1≠UNTIL≠Q≠DO≠YW(/L/).=0.0≠END≠..               SMOOTH
      ≠FOR≠ J.=0 ≠STEP≠ 1 ≠UNTIL≠ NC ≠DO≠                                  SMOOTH
      ≠BEGIN≠ N.=3*J..                                                     SMOOTH
        I.=≠IF≠J≠EQUAL≠ NC ≠THEN≠ 6 ≠ELSE≠ 9..                             SMOOTH
        ≠FOR≠ K.=0•1•2 ≠DO≠                                                SMOOTH
        ≠BEGIN≠ P.=N+K.. O.= 4+K..                                         SMOOTH
          ≠IF≠ KMAX=1 ≠THEN≠                                               SMOOTH
          ≠BEGIN≠≠INTEGER≠T..                                              SMOOTH
            T.=≠IF≠J=NC≠THEN≠2≠ELSE≠5..                                    SMOOTH
            ≠IF≠ EC(/P/) ≠GREATER≠ ≠-10 ≠THEN≠                             SMOOTH
            ≠BEGIN≠ C1.=0.0.. Q.=0..                                       SMOOTH
              ≠FOR≠ L.=0 ≠STEP≠ 1 ≠UNTIL≠ I ≠DO≠                           SMOOTH
              ≠BEGIN≠ R(/P•L/).=0.0.. Q.=Q+1.. C1.=C1+ABS(R1(/P•L/)) ≠END≠.. SMOOTH
              R(/P•O/).=EPSF(/P/).=C1.=(2.0*C1-ABS(R1(/P•O/)))*EPS/Q..     SMOOTH
              C2.=Y(/P/).. YW(/P/).= C1*C2..                               SMOOTH
              Q.=≠IF≠ J=0 ≠THEN≠ 0 ≠ELSE≠ -3..K.=K-1..                     SMOOTH
              ≠FOR≠ L.=Q ≠STEP≠ 1 ≠UNTIL≠ K ≠DO≠                           SMOOTH
              ≠BEGIN≠ O.=N+L..                                             SMOOTH
                M.=≠IF≠ L ≠LESS≠ 0 ≠THEN≠ 8+K ≠ELSE≠ 5+K..                 SMOOTH
                YW(/O/).=YW(/O/)-R(/O•M/)*C2..                             SMOOTH
                R(/O•M/).=0.0..                                            SMOOTH
              ≠END≠.. K.= K+1..                                            SMOOTH
              ≠FOR≠L.=K+1≠STEP≠1≠UNTIL≠T≠DO≠                               SMOOTH
              ≠IF≠EC(/N+L/)≠LESS≠≠-10 ≠THEN≠                               SMOOTH
              YW(/N+L/).=YW(/N+L/)-R1(/P•L+4/)*C2                          SMOOTH
            ≠END≠ ≠ELSE≠                                                   SMOOTH
            ≠BEGIN≠ ≠FOR≠ L.=0 ≠STEP≠ 1 ≠UNTIL≠ I ≠DO≠                     SMOOTH
              R(/P•L/).=R1(/P•L/).. EPSE(/P/).=0..                         SMOOTH
            ≠END≠..                                                        SMOOTH
          ≠END≠ ≠ELSE≠                                                     SMOOTH
          ≠BEGIN≠ EPSE(/P/).=C1.=EPS*EC(/P/)..                             SMOOTH
            ≠FOR≠ L.=0 ≠STEP≠ 1 ≠UNTIL≠ I ≠DO≠ R(/P•L/).= R1(/P•L/)..      SMOOTH
```

```
                R(/P,0/).=R(/P,0/)+C1.. YW(/P/).= C1*Y(/P/).,
              ≠END≠.,
            ≠END≠..
          ≠END≠OF THE DETERMINATION OF H1, AND OF E, S AND EPS..

          ≠BEGIN≠≠COMMENT≠ 5.SOLUTION OF THE EQUATION H1*YCORR=YW, WHERE
               H1=EPS*E+H, YW=EPS*E*Y. THE SOLUTION IS OBTAINED
               USING THE SYMMETRIC CHOLESKI DECOMPOSITION, AND ITERATIVE
               IMPROVEMENT OF Y CORR BY THE RESIDUAL VECTOR METHOD. THE
               MATRIX H1, WHICH IS BLOK-TRIDIAGONAL AND POSITIVE
               DEFINITE, IS DECOMPOSED INTO L*D*L(TRANSP), WHERE D
               IS A DIAGONAL MATRIX AND L A LOWER TRIANGULAR UNIT
               MATRIX. H1 IS STORED IN THE RIGHT HAND SIDE OF THE ARRAY
               P , L IN THE LEFT HAND SIDE. THE UNIT ELEMENTS OF L ARE
               NOT STORED. THE INVERSES OF THE ELEMENTS OF D ARE STORED IN
               ARRAY DC. DURING THE DECOMPOSITION INTERMEDIATE RESULTS
              ARE STORED IN R2..

            ≠REAL≠ ≠PROCEDURE≠ INNERPROD( A,B,C,I,I1,I2,INDEX).,
            ≠VALUE≠ C,I1,I2,INDEX., ≠REAL≠ A,B,C., ≠INTEGER≠ I,I1,I2,INDEX.,
            ≠BEGIN≠ ≠REAL≠ C1,C2,C3,C4,C5,REALMAXINT,NFA,NFB..
              ≠INTEGER≠ M1,M2,M3,N1,N2,N3,R1,R2,R3,R4,S,MAXINT..
              ≠IF≠ INDEX = 1 ≠THEN≠
              ≠BEGIN≠ ≠COMMENT≠ SINGLE PRECISION ACCUMULATION OF PRODUCTS.,
                C1.=C., ≠FOR≠ I.=I1 ≠STEP≠ 1 ≠UNTIL≠ I2 ≠DO≠ C1.=C1+A*B..
                INNERPROD.= C1
              ≠END≠.,
            ≠END≠ OF PROCEDURE INNERPROD.,

            ≠PROCEDURE≠ ALARM(A)., ≠VALUE≠ A., ≠REAL≠ A.,
            ≠BEGIN≠ OUTPUT(41,≠(≠/,≠(≠MATRIX NOT POSITIVE DEFINITE≠)≠,
              2(+ZDB),+D.7D≠+ZZDB≠)≠, J, K, A).,
            ≠END≠ OF ALARM..

            ≠COMMENT≠ 5.1. DECOMPOSITION OF H1 INTO L*D*L(TRANSP)..
            ≠FOR≠J.=0≠STEP≠1≠UNTIL≠NC≠DO≠
            ≠BEGIN≠P.=≠IF≠J=NC≠THEN≠2≠ELSE≠5..
              I.=≠IF≠ J=0 ≠THEN≠ 4 ≠ELSE≠ 1..
              ≠FOR≠ K.=0,1,2 ≠DO≠
              ≠BEGIN≠ N.=3*J+K.,
                ≠FOR≠ L.=K ≠STEP≠ 1 ≠UNTIL≠ P≠DO≠
                ≠BEGIN≠ M.= N-K+L.,
                  Q.=≠IF≠ L ≠LESS≠ 3 ≠THEN≠ K+4 ≠ELSE≠ K+1.,
                  C1.=R(/N,L+4/).,
                  C2.=≠IF≠ L ≠LESS≠ 3 ≠THEN≠
                  -INNERPROD(R(/M,Q/),R2(/N,Q/),-C1,Q,I,3+K,1)
                  ≠ELSE≠(≠IF≠ K=0 ≠THEN≠ C1 ≠ELSE≠
                  -INNERPROD(R(/M,Q/),R2(/N,Q+3/),-C1,Q,1,K,1)).,
                  ≠IF≠L=K≠THEN≠
                  ≠BEGIN≠DC(/N/).=C3.=1.0/C2.,
                   ≠IF≠C2≠LESS≠≠-50≠THEN≠ALARM(C2)
                  ≠END≠ ≠ELSE≠
                  ≠BEGIN≠ R(/M,Q/).=C2*C3.,R2(/M,Q/).=C2 ≠END≠.,
                ≠END≠ OF L-CYCLE.,
              ≠END≠ OF K-CYCLE..
            ≠END≠ OF J CYCLE. AT THIS STAGE H1 HAS BEEN DECOMPOSED INTO
            L*D*L(TRANSP).,
```

```
≠COMMENT≠5.2. COMPUTATION OF Yw..
I.=3*NC+2..
≠FOR≠J.=0≠STEP≠1≠UNTIL≠I≠DO≠ YCORR(/J/).= 0.0..

≠COMMENT≠5.3. BACKSUBSTITUTIONS AND ITERATIVE IMPROVEMENTS..
I.=0..
RETURN..
I.=I+1..C1.=C10.=0.0..C8.=1.0..
≠FOR≠J.=0≠STEP≠1≠UNTIL≠NC≠DO≠
≠BEGIN≠N.=3*J..M.=N-4..
  Q.=≠IF≠J≠EQUAL≠0≠THEN≠4≠ELSE≠1..
  ≠FOR≠K.=0,1,2≠DO≠
  ≠BEGIN≠P.=N+K..O.=3+K..
    YW(/P/).=-INNERPROD(R(/P,L/),YW(/M+L/),-YW(/P/),L,Q,O,1)
  ≠END≠..
≠END≠..
≠FOR≠J.=NC≠STEP≠-1≠UNTIL≠0≠DO≠
≠BEGIN≠ N.=3*J..Q.=N-4..
  M.=≠IF≠J≠EQUAL≠NC≠THEN≠6≠ELSE≠9..
  ≠FOR≠K.=2,1,0≠DO≠
  ≠BEGIN≠P.=N+K..O.=K+1..
    C2.=YW(/P/).=-INNERPROD(R (/Q+L,≠IF≠L≠GREATER≠6≠THEN≠0≠ELSE≠
    K+4/),YW(/Q+L/),-YW(/P/)*DC(/P/),L,K+5,M,1)..
    C3.=YCORR(/P/).=YCORR(/P/)+C2..
    C4.=ABS(C3)..≠IF≠C1≠LESS≠C4≠THEN≠C1.=C4..
    C7.=ABS(C2)..≠IF≠C10≠LESS≠C7≠THEN≠C10.=C7..

    ≠COMMENT≠5.4. THE NEXT SET STATEMENTS CONCERN ACCURACY TESTS..
    ≠IF≠I≠GREATER≠1≠THEN≠
    ≠BEGIN≠C5.=.001/(≠IF≠EC(/P/)≠LESS≠≠-10≠THEN≠≠-20≠ELSE≠EC(/P/))..
      ≠IF≠C5≠LESS≠≠-20≠THEN≠C5.=≠-20.. C6.= C2*C2..
      ≠IF≠C5≠LESS≠≠+7≠AND≠C6≠GREATER≠C5≠THEN≠
      ≠BEGIN≠C8.=-1.0.. ≠IF≠ I=20 ≠THEN≠
        OUTPUT(41,≠(≠/,2(+ZDB),2(+D.7D≠+ZZDB)≠)≠,J,K,C2,C3)..
      ≠END≠ ≠ELSE≠
      ≠IF≠C5≠GREATER≠≠+7≠AND≠C7≠GREATER≠TOL1*C4≠AND≠C7≠GREATER≠TOL2
      ≠THEN≠
      ≠BEGIN≠C8.=-1.0.. ≠IF≠ I=20 ≠THEN≠
        OUTPUT(41,≠(≠/,2(+ZDB),2(+D.7D≠+ZZDB),+ZD≠)≠,J,K,C2,C3,1)..
      ≠END≠..
    ≠END≠ OF I GREATER 1 CONDITIONAL STATEMENT..
  ≠END≠ OF K CYCLE..
≠END≠ OF J CYCLE. AT THIS STAGE YCORR HAS BEEN DETERMINED..

OUTPUT(41,≠(≠/,≠(≠INFINITY NORM OF YCORR    ≠)≠,+D.7D≠+ZZDB≠)≠,C1)..
OUTPUT(41,≠(≠/,≠(≠INFINITY NORM OF IMPROVEMENT VECTOR≠)≠,
+D.7D≠+ZZDB≠)≠,C10)..
≠IF≠I≠GREATER≠1≠AND≠C8≠GREATER≠0.0≠THEN≠≠GOTO≠SOLUTION DETERMINED..
≠IF≠I=1≠THEN≠C8.=-1.0..

≠COMMENT≠5.5.DETERMINATION OF RESIDUALS..
≠FOR≠J.=0≠STEP≠1≠UNTIL≠NC≠DO≠
≠BEGIN≠ N.=3*J..M.=N-4..
  Q.=≠IF≠J≠EQUAL≠0≠THEN≠4≠ELSE≠1..
  O.=≠IF≠J≠EQUAL≠NC≠THEN≠6≠ELSE≠9..
  ≠FOR≠K.=0,1,2≠DO≠
  ≠BEGIN≠P.=N+K..
    ≠IF≠KMAX=1≠THEN≠
```

```
           ≠BEGIN≠≠INTEGER≠R.,

             ≠REAL≠≠PROCEDURE≠ELEMENT OF R1(L)., ≠INTEGER≠ L.,
             ELEMENT OF R1.=≠IF≠L≠LESS≠4≠THEN≠
             R1(/M+L,K+7/)≠ELSE≠(≠IF≠L≠LESS≠K+4≠THEN≠
             R1(/M+L,K+4/)≠ELSE≠R1(/P,L/)).,

             YW(/P/).=-INNERPROD(YCORR(/M+L/),
             ≠IF≠EC(/P/)≠GREATER≠≠-100≠OR≠EC(/M+L/)≠GREATER≠≠-100
             ≠THEN≠ 0.0 ≠ELSE≠ ELEMENT OF R1(L),
             INNERPROD(YCORR(/M+R/),≠IF≠EC(/P/)≠LESS≠≠-100≠AND≠
             EC(/M+R/)≠GREATER≠≠-100≠THEN≠ELEMENT OF R1(R)≠ELSE≠0.0,
             -EPSE(/P/)*(Y(/P/)-YCORR(/P/)),R,Q,0,1),L,Q,0,1)
           ≠END≠≠ELSE≠
           YW(/P/).=-INNERPROD(YCORR(/M+L/),
           ≠IF≠ L ≠LESS≠ 4 ≠THEN≠ R1(/M+L,K+7/)
           ≠ELSE≠(≠IF≠L≠LESS≠4+K≠THEN≠R1(/M+L,4+K/)≠ELSE≠R1(/P,L/)),
           -EPSE(/P/)*(Y(/P/)-YCORR(/P/)),L,Q,0,1)
         ≠END≠
        ≠END≠ AT THIS STAGE THE RESIDUALS HAVE BEEN STORED IN YW.,
        ≠IF≠ C8 ≠LESS≠ 0 ≠AND≠ I ≠LESS≠ 7 ≠THEN≠ ≠GOTO≠ RETURN.,
      ≠END≠ OF SOLUTION OF THE EQUATION H1*YCORR=YW.,
    SOLUTION DETERMINED..
      KC.=KC+1., ≠GOTO≠AGAIN.,
    READY.. KC.=KC.,
      ≠IF≠ KKC ≠LESS≠ KKMAX ≠THEN≠ ≠GOTO≠ ONCE MORE 1.,
    ≠END≠ OF THE PROCEDURE SMOOTH THE VECTOR Y.,

    ≠PROCEDURE≠ INTERPOLATE(X,K0,K1,K2,K3,K4,K5)., ≠VALUE≠ X.,
    ≠REAL≠ X., ≠INTEGER≠ K0,K1,K2,K3,K4,K5.,
    ≠BEGIN≠ ≠REAL≠ XJ,XJPE,XL,XR,DX,C1,C2,C3,SUM.,
      ≠INTEGER≠ I,J,K,L.,
      ≠BOOLEAN≠ A COMPUTED.,
      ≠REAL≠ ≠ARRAY≠ A1(/1..6,3..5/),DCX(/1..5/).,
      A COMPUTED.=IC≠GREATER≠0.,
      ≠IF≠ IC≠NOTGREATER≠0 ≠THEN≠ IC.=1.,
      XJPE.=XC(/IC/)., XJ.=XC(/IC-1/).,
      ≠COMMENT≠ 1. DETERMINATION OF INTERVAL CONTAINING X.,
      INTERVAL..
      ≠IF≠ XJPE-XJ≠GREATER≠0≠THEN≠
      ≠BEGIN≠ ≠IF≠ XJPE≠GREATER≠X≠AND≠XJ≠NOTGREATER≠X≠THEN≠≠GOTO≠PROCEED
      ≠END≠ ≠ELSE≠
      ≠BEGIN≠ ≠IF≠ XJPE≠LESS≠X≠AND≠XJ≠NOTLESS≠X≠THEN≠ ≠GOTO≠ PROCEED
      ≠END≠.,
      ≠IF≠ IC≠EQUAL≠ NC ≠THEN≠
      ≠BEGIN≠ IC.=1., XJ.=XC(/0/).,
        XJPE.=XC(/1/)
      ≠END≠ ≠ELSE≠
      ≠BEGIN≠ IC.=IC+1., XJ.=XJPE.,XJPE.=XC(/IC/)
      ≠END≠.,
      A COMPUTED.= ≠FALSE≠., ≠GOTO≠ INTERVAL.,

      ≠COMMENT≠ 2. COMPUTATION OF ARRAY A1 AND OF A3,A4,A5.,
      PROCEED..
      ≠IF≠ ≠NOT≠ A COMPUTED ≠THEN≠
      ≠BEGIN≠ DX.=XJPE-XJ., C1.=C2.=1.0/DX.,
        ≠FOR≠ I.=1 ≠STEP≠ 1 ≠UNTIL≠ 5 ≠DO≠
        ≠BEGIN≠ DCX(/I/).=C2., C2.= C2*C1
```

```
        ≠END≠.,
        K.=3*IC-4.,
        ≠FOR≠ I.=3,4,5 ≠DO≠
        ≠BEGIN≠ A1(/4,I/).=C3.=(2*I-5+(≠IF≠I≠EQUAL≠5≠THEN≠1≠ELSE≠0))*
          (-1.0)≠POWER≠(I-1)*DCX(/I/).,
          A1(/1,I/).=-C3.,
          A1(/2,I/).=C2.=(I-2)*(-1.0)≠POWER≠(I-2)*DCX(/I-1/).,
          A1(/5,I/).=-(C3*DX+C2).,
          A1(/3,I/).=0.5*(-1.0)≠POWER≠(I-2)*DCX(/I-2/).,
          A1(/6,I/).=(I-3)*(I-4)*0.25*DCX(/3/)
        ≠END≠.,
        ≠FOR≠ I.=3,4,5 ≠DO≠
        ≠BEGIN≠ SUM.=0.,
          ≠FOR≠ L.=1 ≠STEP≠ 1 ≠UNTIL≠ 6 ≠DO≠
          SUM.=SUM+A1(/L,I/)*YCORR(/L+K/).,
          ≠IF≠ I≠EQUAL≠3 ≠THEN≠ A3.=SUM ≠ELSE≠ ≠IF≠ I≠EQUAL≠4 ≠THEN≠
          A4.=SUM ≠ELSE≠ A5.=SUM.,
        ≠END≠.,
      ≠END≠.,

      ≠COMMENT≠ 3. COMPUTATION OF INTERPOLATED VALUES.,
      XL.=X-XJ., XR.=X-XJPE.,K.=3*IC-4.,
      ≠IF≠ K0≠EQUAL≠0 ≠THEN≠
      Y0.=YCORR(/K+1/)+XL*(YCORR(/K+2/)+XL*(0.5*YCORR(/K+3/)+
      XL*(A3+XR*(A4+XR*A5)))).,
      ≠IF≠ K1≠EQUAL≠1 ≠THEN≠
      Y1.=YCORR(/K+2/)+XL*(YCORR(/K+3/)+XL*
      (3.0*(A3+XR*(A4+XR*A5))+XL*(A4+XR*2.0*A5))).,
      ≠IF≠ K2≠EQUAL≠2 ≠THEN≠ Y2.=YCORR(/K+3/)+XL*(6.0*
      (A3+XR*(A4+XR*A5))+XL*(6.0*(A4+2.0*XR*A5)+XL*2.0*A5)).,
      ≠IF≠ K3≠EQUAL≠3 ≠THEN≠
      Y3.=6.0*(A3+XL*(3.0*A4+XR*6.0*A5+XL*(3.0*A5))+XR*(A4+XR*A5)).,
      ≠IF≠ K4≠EQUAL≠4 ≠THEN≠
      Y4.=24.0*(A4+(2.0*XR+3.0*XL)*A5).,
      ≠IF≠ K5≠EQUAL≠5 ≠THEN≠ Y5.=120*A5.,
    ≠END≠ OF PROCEDURE INTERPOLATE.,

    ≠COMMENT≠ MAIN PROGRAM.,

    OUTPUT(41,≠(≠*,≠(≠RESULTS OF PROGRAM T 32≠)≠,/,
    ≠(≠DATA INPUT TAPE NUMBER A≠)≠,+ZD≠)≠,IC).,
    ≠FOR≠ I.=0 ≠STEP≠ 1 ≠UNTIL≠ NC ≠DO≠
    ≠BEGIN≠ J.=3*I.,
      INPUT(40,≠(≠≠)≠,XC(/I/),Y(/J/),Y(/J+1/),Y(/J+2/)).,
      OUTPUT(41,≠(≠/,+ZDB,4(+D.7D≠+ZDB)≠)≠,
      I,XC(/I/),Y(/J/),Y(/J+1/),Y(/J+2/)).,
    ≠END≠.,
    OUTPUT(41,≠(≠*,≠(≠MODIFIED INPUT DATA≠)≠≠)≠).,
    ≠FOR≠ I.=0 ≠STEP≠ 1 ≠UNTIL≠ NC ≠DO≠
    ≠BEGIN≠ J.=3*I.,
      C1.=COS(Y(/J+1/)).,
      Y(/J+1/).=SIN(Y(/J+1/))/C1.,
      Y(/J+2/).=-Y(/J+2/)/C1≠POWER≠3.,
      OUTPUT(41,≠(≠/,+ZDB,4(+D.7D≠+ZZDB)≠)≠,
      I, XC(/I/), Y(/J/), Y(/J+1/), Y(/J+2/)).,
    ≠END≠.,

    SMOOTH THE VECTOR Y.,
```

```
        OUTPUT(41,≠(≠*,≠(≠RESULTS OF THE SMOOTHING PROCESS≠)≠≠)≠).,
        ≠FOR≠ I.=0 ≠STEP≠ 1 ≠UNTIL≠ NC ≠DO≠
        ≠BEGIN≠ J.=3*I., OUTPUT(41,≠(≠/,+ZDB,7(+D.7D≠+ZZDB)≠)≠,
          I,XC(/I/),YCORR(/J/),YCORR(/J+1/),YCORR(/J+2/),
          YCORR(/J/)-Y(/J/),YCORR(/J+1/)-Y(/J+1/),YCORR(/J+2/)-Y(/J+2/)).,
        ≠END≠.,
  470**
        OUTPUT(41,≠(≠*,≠(≠RESULTS OF INTERPOLATION≠)≠≠)≠).,
        IC.=0.,
        ≠FOR≠ I.=0 ≠STEP≠ 1 ≠UNTIL≠ NC-1 ≠DO≠
        ≠BEGIN≠ C1.=(XC(/I+1/)-XC(/I/))/6.0.,
          ≠FOR≠ J.=0 ≠STEP≠ 1 ≠UNTIL≠ 6 ≠DO≠
          ≠BEGIN≠ C2.= ≠IF≠ J ≠LESS≠ 6 ≠THEN≠ XC(/I/)+J*C1 ≠ELSE≠
            XC(/I+1/)*(1.0-SIGN(C1)*SIGN(XC(/I+1/))*≠-10).,
            OUTPUT(41,≠(≠B≠)≠)., INTERPOLATE(C2,0,1,2,3,4,5).,
            OUTPUT(41,≠(≠/,+ZDB,7(+D.7D≠+ZZDB)≠)≠,I,C2,Y0,Y1,Y2,Y3,Y4,Y5).,
  480**   ≠END≠
        ≠END≠.,

        OUTPUT(41,≠(≠*,≠(≠CORRECTED AEROFOIL SECTION≠)≠,/,
        ≠(≠I, NEW VALUES, OLD VALUES MINUS NEW VALUES≠)≠≠)≠).,
        ≠FOR≠ I.=0 ≠STEP≠ 1 ≠UNTIL≠ NC ≠DO≠
        ≠BEGIN≠ J.=3*I.,
          OUTPUT(41,≠(≠/,+ZDB,7(+D.7D≠+ZZDB)≠)≠,
          I, XC(/I/), YCORR(/J/), ARCTAN(YCORR(/J+1/)),
          -YCORR(/J+2/)/((1.0+YCORR(/J+1/)≠POWER≠2)≠POWER≠1.5),
  490**   Y(/J/)-YCORR(/J/),
          ARCTAN(Y(/J+1/))-ARCTAN(YCORR(/J+1/)),
          -Y(/J+2/)/((1.0+Y(/J+1/)≠POWER≠2)≠POWER≠1.5)
          +YCORR(/J+2/)/((1.0+YCORR(/J+1/)≠POWER≠2)≠POWER≠1.5)).,
        ≠END≠.,
          IC.=0.,
        OUTPUT(41,≠(≠//,≠(≠ END OF RESULTS OF PROGRAM T 32 ≠)≠≠)≠).,
        ≠GOTO≠ START OF PROGRAM.,
        ≠END≠ OF BLOK 2.,

  500** RUNS COMPLETED.. NC.=NC.,
        ≠END≠
                ≠EOP≠
```

```
LINE    0     PROGRAM BEGINS          (MESSAGE)     1
LINE  501     PROGRAM ENDS            (MESSAGE)     1
LINE  501     SOURCE DECK ENDS        (MESSAGE)     1
LINE   48     NON-FORMAT STRING       (MESSAGE)     1
LINE   48     NON-FORMAT STRING       (MESSAGE)     1
LINE   48     NON-FORMAT STRING       (MESSAGE)     1
LINE   48     NON-FORMAT STRING       (MESSAGE)     1
LINE   49     NON-FORMAT STRING       (MESSAGE)     1
LINE   49     NON-FORMAT STRING       (MESSAGE)     1
LINE   49     NON-FORMAT STRING       (MESSAGE)     1
LINE   51     NON-FORMAT STRING       (MESSAGE)     1
LINE   51     NON-FORMAT STRING       (MESSAGE)     1
LINE   51     NON-FORMAT STRING       (MESSAGE)     1
LINE   53     NON-FORMAT STRING       (MESSAGE)     1
LINE   53     NON-FORMAT STRING       (MESSAGE)     1

      THE FOLLOWING CONTROL CARD OPTIONS ARE ACTIVE     F,I,L
```